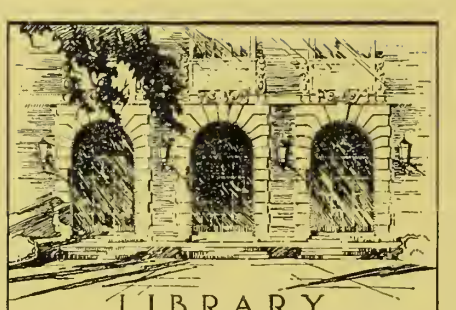


LIBRARY
OF THE
UNIVERSITY
OF ILLINOIS

510.84
Il6r
no.140-147
cop.2

CENTRAL CIRCULATION BOOKSTACKS

The person charging this material is responsible for its renewal or its return to the library from which it was borrowed on or before the **Latest Date** stamped below. **You may be charged a minimum fee of $75.00 for each lost book.**

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

TO RENEW CALL TELEPHONE CENTER, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

OCT 05 1996
NOV 20 1996

When renewing by phone, write new due date below previous due date. L162


Digitized by the Internet Archive
in 2013

http://archive.org/details/cyclicspoterrorc141foul

510.84
Il6r
no.141
cop. 2

UNIVERSITY OF ILLINOIS
GRADUATE COLLEGE
DIGITAL COMPUTER LABORATORY

REPORT NO. 141

CYCLIC SPOT-ERROR-CORRECTING CODES

by

Clinton R. Foulk

July 22, 1963

(This work is being submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Mathematics, July, 1963, and was supported in part by the Office of Naval Research under Contract Nonr-1834(27).)

UNIVERSITY OF ILLINOIS
GRADUATE COLLEGE
DIGITAL COMPUTER LABORATORY

REPORT NO. 141

CYCLIC SPOT-ERROR-CORRECTING CODES

by

Clinton R. Foulk

July 22, 1963

(This work is being submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Mathematics, July, 1963, and was supported in part by the Office of Naval Research under Contract Nonr-1834(27).)

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Professor David E. Muller for his indispensable guidance and assistance in the preparation of this thesis, and to his wife, Dorothy, for her patience, forbearance, and encouragement in its preparation.

TABLE OF CONTENTS

	Page
CHAPTER 0. INTRODUCTION	1
CHAPTER I. ERROR-CORRECTING CODES	7
Section I-1. Cyclic Group Codes	7
Section I-2. Binary Code-Generating Polynomials	19
Section I-3. Factorable Code-Generating Polynomials	27
Section I-4. Efficiency	32
CHAPTER II. CODES DEFINED ON RASTERS	35
Section II-1. Correction of Cyclic Spot-Errors Defined on a Raster	35
Section II-2. Correction of Cyclic Spot-Errors on Rasters Determined by Matrix Pairs	42
Section II-3. Methods for Constructing a Cyclic Spot-Error-Correcting Code on a Given Raster	55
CHAPTER III. ONE-DIMENSIONAL CODES	60
Section III-1. The Fire Method	60
Section III-2. The Modified Combination Method	61
Section III-3. The Computer Search	68
Section III-4. Summary of One-Dimensional Results	75
CHAPTER IV. TWO- AND THREE-DIMENSIONAL CODES	83
Section IV-1. Diameter of a Two-Dimensional Raster	83
Section IV-2. The Direct Method	88
Section IV-3. The Fire Method	100
Section IV-4. The Combination Method	101
Section IV-5. Comparison with Iterated Two-Dimensional Codes	103
BIBLIOGRAPHY	105
VITA	106

CHAPTER 0

INTRODUCTION

The problem of transmitting and storing large amounts of information with high reliability has recently come under investigation. By "transmission" of information, we refer to any process which first stores a message in some storage medium or information channel, such as, for example, radio waves or magnetic tape, and then retrieves this message for use at some other time or place. Instead of working to make physical components absolutely reliable, it is possible to transmit a certain amount of redundant information with each message so that if some component fails during transmission, the message can be corrected at the receiver. In order to do this, there must of course be some agreement between the transmitter and the receiver as to what form this redundant information is to take, just as there must originally be some agreement as to what form the message itself is to take.

Several methods of adding redundant information to a message have been suggested. One method is to have the receiver transmit each received message back to the transmitter where it is compared with the original message to see if it is correct. Another method is to transmit the whole message three times and take a "majority vote" among each of the received symbols to ascertain what they should be.

The method which we shall investigate is an outgrowth of the "parity check". A parity check bit is a single zero or one bit of information added to a binary message so as to make the number of one's in the entire message be of odd or even parity, whichever has been agreed upon. Then, if an odd number of errors occurs in the transmission of a message, this fact is immediately detected by the receiver because the number of ones is of the wrong parity.

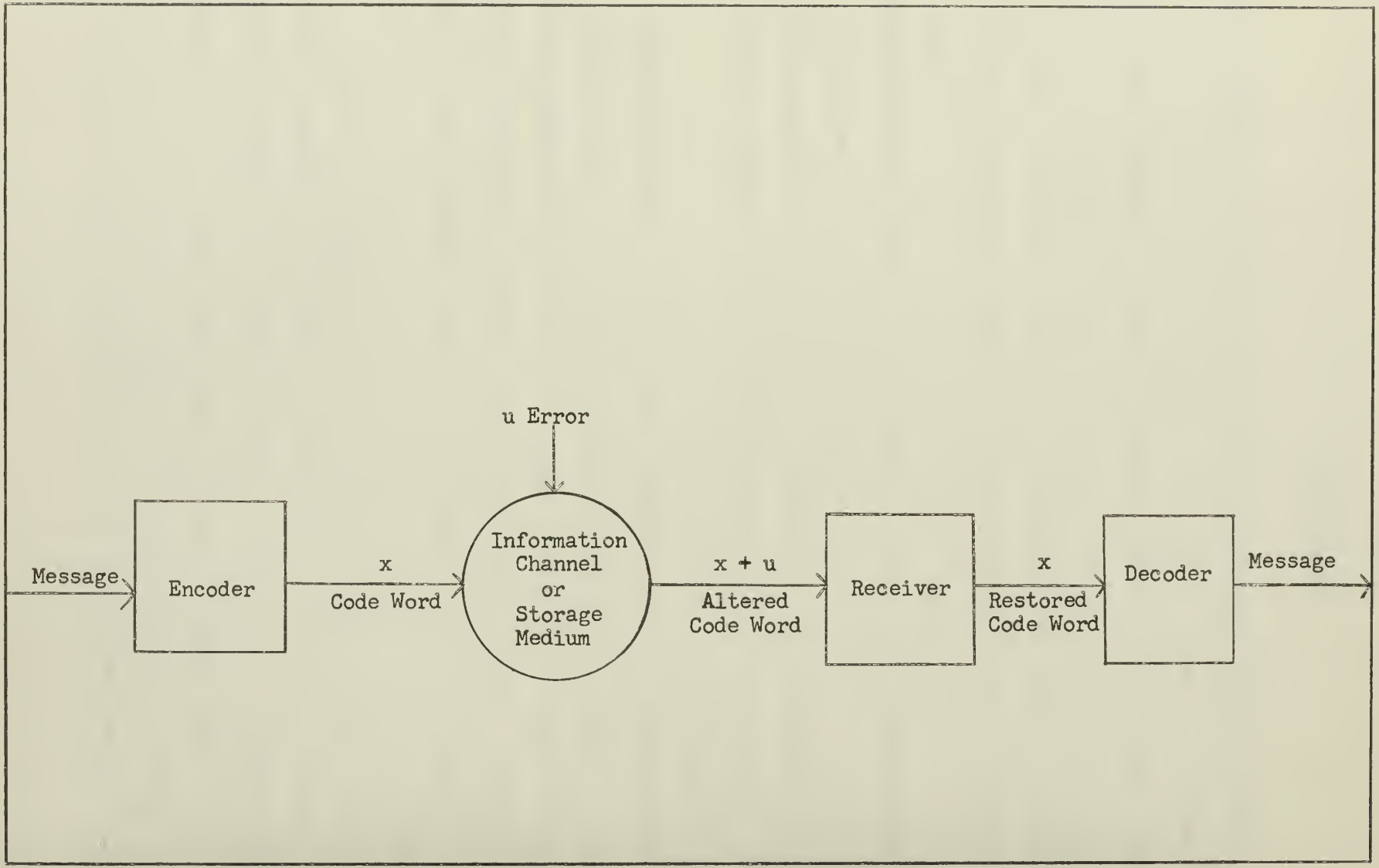

u Error
Message
Encoder
x
Code Word
Information
Channel
or
Storage
Medium
x + u
Altered
Code Word
Receiver
x
Restored
Code Word
Decoder
Message

Figure 0-1

Hamming was the first to demonstrate that, by transmitting multiple parity check bits with each message, it is possible not only to detect that an error has occurred in transmission, but actually to correct an error if it is one in some prearranged set of errors.

Figure 0-1 illustrates how an error-correcting code is used to transmit a message. We imagine that we have a set of messages which we wish to send from the transmitter to the receiver through an information channel or storage medium. The encoder first selects one of the permissible code words to represent the given message and inserts it into the information channel or storage medium. We imagine that while the code word is in this channel or medium, it is subject to certain errors which manifest themselves as alterations in the code word. Therefore, the received n-tuple must either be itself a code word, or it must consist of the code word originally selected by the encoder altered by some permissible error. The receiver reconstructs the original code word from the received n-tuple, and the decoder determines which message is represented by the code word. It is obvious that in order for this method to work, each of the n-tuples obtained from the permissible alterations of a given code word must be different from any n-tuple obtained from a permissible alteration of any other code word. (This is the condition for correction described in Section I-1.)

There are a great many advantages which result from the use of a group code, that is, a code which is closed under the operations of vector addition and scalar multiplication; both in the mathematical analysis of the code and in its practical implementation. One of the most important mathematical advantages of the group code is that the set of all code words of a given code is precisely the null space of some matrix A. Still further advantages result from the use of a cyclic group code, for the set of all code words of a given cyclic group code of length n is precisely the set of all multiples of de-

gree $\leq n - 1$ of some polynomial P(t). In the case of a cyclic group code, therefore, it is only necessary to divide each n-tuple arriving at the receiver by P(t); if the remainder of this division is zero, we know that this n-tuple is a code word and therefore, by the condition for correction, it is the code word transmitted originally. If, on the other hand, this remainder is non-zero, then some error has occurred in the transmission of the code, and, in order to satisfy the condition for correction, there must be a distinct remainder for each of the permissible errors.

There are a number of methods which can be used for the physical realization of the process of deducing the correct code word and the error from a given received n-tuple. Peterson, p. 193, gives a method which takes advantage of the cyclic group properties of the code by using a feedback shift register realizing P(t) and a buffer storage register, and thus requires much less hardware than a simple matrix logic decoder.

The reader who is familiar with the contents of Peterson's Error-Correcting Codes should have no difficulty with this paper, as we assume a familiarity with error-correcting codes approximately equal to that provided by his book.

In Section I-1, we give a formal definition of codes and sets of errors in general, then proceed to define group codes, and finally cyclic group codes. The group code has the advantage, as we pointed out, that every element of the code is in the null space of a matrix A and therefore we can use the known results of matrix theory to expedite our investigation of these codes. Furthermore, the cyclic group codes have the property that every element in the code is some multiple of a generating polynomial, which means we can use the known results concerning polynomials with coefficients in a Galois Field to expedite our investigation of these codes. Finally, consideration is given to the problem of defining the efficiency of a cyclic group code.

Beginning with Section I-2, we restrict our investigations to the binary case. We first consider the cyclic structure of polynomials with coefficients in GF(2) with a view to using the results obtained to find cyclic group codes for transmitting binary information. This discussion also gives us an insight into the method by which a given polynomial generates an error-correcting code and a method for determining whether or not a given polynomial generates a given cyclic group code.

In Section I-3, it is shown that many of the error-correcting properties of the code generated by a product polynomial can be predicted from the cyclic structures of its factors. This enables us to use polynomials of low degree, whose error-correcting properties can be determined with relatively little effort, to construct polynomials with similar error-correcting properties but of high degree. The resulting factorable polynomials usually generate less efficient codes than irreducible polynomials of comparable degree, but are much easier to find since the detailed properties of cyclic structures of irreducible polynomials are to a large extent unpredictable.

In Section I-4, we give a more complete discussion of the concept of efficiency.

In Section II-1, we first define what we mean by a raster and spot, what we mean when we say that a code is defined on a raster, and how the results of Section I-3 on factorable polynomials can be applied to rasters appropriately related by their respective distance functions.

In Section II-2, a distance function is defined which is based on the ordinary Euclidean distance function and yet satisfies the cyclic hypothesis of Section II-1. The raster itself is defined as a set of equivalence classes of a particular set of points in Euclidean space. The diameter of a raster is defined and the application of the results of Section II-1 to such rasters is discussed.

In Section II-3, we discuss methods of actually constructing codes on given rasters by each of three different methods; the Direct Method, the Fire Method, and the Combination Method. We also discuss the methods by which a given raster and code can be "expanded", thus greatly increasing the usefulness of the results given in Chapters III and IV.

In Chapter III, we show how to construct codes defined on one-dimensional rasters and give a table of polynomials generating one-dimensional error-correcting codes. These are actually arrived at by a special method which insures that the codes obtained are always maximally efficient.

In Section IV-1, we discuss the properties of typical two-dimensional rasters and codes defined on them, and give examples of the construction of a larger raster from a given one.

In Section IV-2, we give tables showing various two-dimensional error-correcting codes obtained by the Direct Method and give examples which show exactly how these results are arrived at.

In Section IV-3, we show how the Fire Method can be used to find spot-error-correcting codes in two or more dimensions.

In Section IV-4, we show how to combine the Direct Method and the Fire Method to give the Combination Method of constructing spot-error-correcting codes of high efficiency.

Finally, in Section IV-5, we compare our codes with the iterated two-dimensional codes investigated by Elspas.

CHAPTER I

ERROR-CORRECTING CODES

Section I-1. Cyclic Group Codes.

Definition I-1-1: A code C of length n is a non-empty subset of the set of ordered n-tuples $x = (x_0, x_1, \ldots, x_{n-1})$, where each coordinate x_i is a symbol from some finite alphabet F.

We shall consider only the case in which the alphabet is a finite field $F = GF(q)$, where $q = p^k$ for some prime p and integer k; C is then a set of vectors in F^n.

Definition I-1-2: A code word is any n-tuple in a code C.

Definition I-1-3: An error function E is a mapping defined on the elements x in C such that the image E(x) of each x in C is some subset of F^n.

Definition I-1-4: An error function E satisfies the condition for correction iff $E(x) \cap E(y) = \emptyset$ whenever x and y are in C and $x \neq y$.

Definition I-1-5: Let E be an error function defined by means of a set D of ordered n-tuples of F^n and the operation of vector addition as follows: for each x in C, E(x) is the set of all n-tuples of the form $x + u$, where u is in D. Then D is said to be a set of errors defining E.

Definition I-1-6: An error is any n-tuple in a set of errors D.

Definition I-1-7: The code C corrects the set of errors D iff the error function E defined by D satisfies the condition for correction.

Theorem I-1-1: If C is a code correcting a set of errors D and N(C) and N(D) are the numbers of elements in C and D respectively, then $N(C) \times N(D) \leq q^n$.

Proof: Since the error function E(x) defined by D satisfies the condition for correction, we know that there exist N(D) distinct n-tuples corresponding to each of the N(C) distinct code words in C. However, there exist only q^n distinct n-tuples with coefficients in GF(q); therefore, we see that

$N(C) \times N(D) \leq q^n$.

This theorem establishes an information-theoretical upper bound on the product of the number of n-tuples in C by the number of n-tuples in D which depends only on n and q.

It should be noted that C and D are not necessarily disjoint and that the sum of the number of n-tuples in C plus the number of n-tuples in D may be greater than q^n. If, however, the field of coefficients is the binary field GF(2), then C and D can have at most one element in common, since the sum of any binary n-tuple plus itself is always the zero n-tuple. Thus, in the binary case, the condition for correction would be violated if C and D had more than one element in common. Usually this single element will be the zero n-tuple 0, for if C is a group code (see Definition I-1-10), then 0 must be in C; at the same time, we shall almost always wish to assume that it is possible for a code word to be transmitted unimpaired; i.e., that D contains the error 0.

It should also be noted that it is possible to have a code C which contains only the zero element while D contains all q^n n-tuples; or, on the other hand, to have a code C which contains all q^n n-tuples while D contains only the zero element. In the first case, we would be able to transmit only the zero code word but would have correction of all q^n possible errors; in the second case we would be able to transmit all q^n possible code words but would have correction of only the zero error. These extreme examples are of little interest in themselves, but are very important in the construction of codes by the Fire Method described in Sections II-3 and IV-3.

Theorem I-1-2: The code C corrects the set of errors D iff $x + u = y + v$ implies $x = y$ and $u = v$ whenever x and y are in C and u and v are in D.

Proof: First, suppose that $x + u = y + v$ implies $x = y$ and $u = v$ whenever x and y are in C and u and v are in D. Then for x and y in C and $x \neq y$, we

have $x + u \neq y + v$ for every u and v in D. This means that $E(x) \cap E(y) = \emptyset$ whenever x and y are in C and $x \neq y$; therefore, C corrects D.

Conversely, suppose that C corrects the set of errors D. Then $E(x) \cap E(y) = \emptyset$ whenever x and y are in C and $x \neq y$. Since E(x) is the set of all n-tuples of the form $x + u$ with u in D and similarly E(y) is the set of all n-tuples $y + v$ with v in D, we see that $x + u \neq y + v$ whenever x and y are in C, u and v are in D, and $x \neq y$. Therefore, $x + u = y + v$ with x and y in C and u and v in D implies that $x = y$ which in turn implies that $u = v$. Therefore, $x + u = y + v$ implies $x = y$ and $u = v$ whenever x and y are in C and u and v are in D.

Definition I-1-8: An error function E is said to be close-packed iff for each y in F^n there exists an x in C such that y is in E(x).

Definition I-1-9: A code C which corrects a set of errors D is said to be close-packed with respect to D iff the error function E defined by D is close-packed.

Theorem I-1-3: A code C which corrects a set of errors D is close-packed with respect to D iff each vector in F^n is uniquely expressible as a sum $x + u$, where x is in C and u is in D.

Proof: Let y be any vector in F^n. C is close-packed with respect to D iff there exists an x in C such that y is in E(x), where E(x) is the error function defined by D; that is, iff there is a u in C such that $y = x + u$. By Theorem I-1-2, x and u are unique.

Corollary: If a code C is close-packed with respect to a set of errors D corrected by C, then $N(C) \times N(D) = q^n$.

For a further discussion of the concept of "close-packed", see Section I-4.

Definition I-1-10: A group code C is a code which is also a vector space; i.e., a code which is closed under the operations of component-wise addition and scalar multiplication.

Theorem I-1-4: A code C is a group code iff there is a matrix A of rank $m < n$ with coefficients in F such that $Ax = 0$ iff $x \in C$. (Note that although A is not unique all matrices A have the same rank m.)

Proof: This is a well-known theorem of matrix algebra.

Definition I-1-11: If C is a group code defined by $C = \{x | Ax = 0\}$, then the matrix A is said to determine C, and the rank m is said to be the number associated with C.

Theorem I-1-5: If A is an m x n matrix of rank m which determines a group code C, then C corrects a set of errors D iff all m-tuples Au are distinct as u ranges over D.

Proof: First, suppose x and y are in C and u and v are in D and that all m-tuples Au are distinct as u ranges over D. This means that $Au = Av$ implies that $u = v$. By the rules of matrix multiplication, $x + u = y + v$ implies $A(x + u) = A(y + v)$, which implies that $Au = Av$, which implies that $u = v$, and this finally implies that $x = y$. Summarizing, we see that $x + u = y + v$ implies that $x = y$ and $u = v$, so C corrects D.

Conversely, suppose that C corrects D. We shall prove that $Au = Av$ implies that $u = v$. To this end, assume that $Au = Av$ and let $u - v = w$. Then w is in C because $Aw = A(u - v) = Au - Av = 0$. Then $0 + u = w + v$ implies that $w = 0$ and $u = v$ because 0 and w are in C and u and v are in D.

Theorem I-1-6: If A is an m x n matrix of rank m which determines a group code C correcting a set of errors D, then the pair (C,D) is close-packed iff Au generates all m-vectors as u ranges over D.

Proof: First, suppose that the pair (C,D) is close-packed, and let z be an arbitrary m-vector. Since A is m x n and of rank m, we may solve Aw = z for some n-tuple w. By hypothesis, w is uniquely expressible in the form w = x + u where x is in C and u is in D. Therefore, the arbitrary m-vector z is generated by the element u in D since z = Aw = A(x + u) = Au.

Conversely, suppose that Au generates all m-tuples z as u ranges over D. Let w be any n vector and suppose that Aw = z. By hypothesis, there exists a u in D such that Au = z, which implies that Aw = Au and therefore that A(w - u) = Ax = 0; therefore w - u = x is in C. Therefore, we have w = x + u, where x is in C and u is in D.

Corollary: If A is an m x n matrix of rank m which determines a group code C correcting a set of errors D, then the pair (C,D) is close-packed iff there is a one-to-one correspondence between the set of all m-vectors and the set of elements u in D.

Theorem I-1-7: If A is an m x n matrix of rank m which determines a group code C correcting a set of errors D, and the number of elements in F is q, then the pair (C,D) is close packed iff the number of elements in D is q^m.

Proof: First, suppose that the pair (C,D) is close-packed. Then, because of the one-to-one correspondence between elements of D and the set of all m-vectors, we see that D contains exactly q^m elements.

Conversely, suppose that D contains q^m elements. By Theorem I-1-5, all m-vectors are distinct as u ranges over D; therefore Au generates all q^m m-vectors as u ranges over D.

Definition I-1-12: A group code C which corrects a set of errors D is said to be efficient with respect to D iff $N(C) \times N(D) > q^{n-1}$.

Theorem I-1-8: If a group code C exists which corrects a set of errors D and is efficient with respect to D, then no group code $\hat{C}$ exists which also corrects D and contains a larger number of elements than C.

Proof: Since C and $\hat{C}$ are both group codes, N(C) and $N(\hat{C})$ are both powers of q. Let $N(C) = q^k$; then $N(\hat{C}) \geq q^{k+1}$. Since by hypothesis $N(C) \times N(D) = q^k N(D) > q^{n-1}$, we see that $N(\hat{C}) \times N(D) \geq q^{k+1} N(D) > q^n$, which is impossible by Theorem I-1-1. Therefore, no such group code $\hat{C}$ exists.

For a further discussion of the concept of "efficient", see Section I-4.

Definition I-1-13: A group code C is cyclic iff $(x_0, x_1, \ldots, x_{n-1}) = x$ in C implies that $(x_{n-1}, x_0, \ldots, x_{n-2}) = x'$ is in C, and similarly a set of errors D is cyclic iff $(u_0, u_1 \ldots, u_{n-1}) = u$ in D implies that $(u_{n-1}, u_0, \ldots, u_{n-2}) = u'$ is in D.

Definition I-1-14: We define $P_x(t) = x_0 + x_1 t + \ldots + x_{n-2} t^{n-2} + x_{n-1} t^{n-1}$, and say that $P_x(t)$ is in C* iff x is in C and, similarly, $P_u(t)$ is in D* iff u is in D.

The symbol t introduced in Definition I-1-14 is an indeterminate which can be added and multiplied just as though it were an element of GF(q), and therefore generates a ring of polynomials with coefficients in GF(q). The reason for introducing such an indeterminate is that, as Theorem I-1-10 shows, there is a one-to-one correspondence between cyclic group codes of length n and monic polynomials P(t) which divide $t^n - 1$.

Theorem I-1-9: If C is a cyclic group code determined by an m x n matrix A of rank m, then there exists a monic polynomial P(t) of degree m with non-zero constant term such that $P_x(t)$ is in C* iff $P_x(t)$ can be expressed in the form $P_x(t) = R(t)P(t)$, where (degree of R(t)) + (degree of P(t)) $\leq n - 1$.

Proof: Since C is a cyclic group code, $P_x(t)$ in C* and $P_y(t)$ in C* imply that $P_x(t) + P_y(t) = P_{x+y}(t)$ is in C*; a in F and $P_x(t)$ in C* imply that

$aP_x(t) = P_{ax}(t)$ is in C*; and $P_x(t)$ in C* implies that $P_{x'}(t) = tP_x(t) - x_{n-1}(t^n - 1)$ is in C*. In other words, $P_{x'}(t) = \text{Res } tP_x(t) (\text{mod } t^n - 1)$ is in C*. This means that $P_{x^{(j)}}(t) = \text{Res } t^j P_x(t) (\text{mod } t^n - 1)$ is in C* for any integer j. Finally, $\text{Res } A(t)P_x(t) (\text{mod } t^n - 1)$ is in C* for any polynomial A(t). Using the Euclidean algorithm, we first find the monic polynomial P(t) of maximum degree m* which divides all $P_x(t)$ in C*. This polynomial is necessarily unique and has a non-zero constant term since C* contains polynomials which are not divisible by t. Now, we can find two sets of polynomials, $P_i(t)$ in C* and $A_i(t)$, such that $\sum_i A_i(t)P_i(t) = P(t)$; therefore P(t) is in C*.

Since P(t) is the g.c.d. of all $P_x(t)$ in C*, every $P_x(t)$ in C* is of the form R(t)P(t), where R(t) is some polynomial of degree $\leq n - m^* - 1$.

Since P(t) is in C* and $A(t)P_x(t)$ is in C* for all A(t) and x, the polynomial R(t)P(t) is in C* for every R(t) of degree $\leq n - m^* - 1$.

The fact that $m^* = m$ will become apparent after we have come to Theorems I-1-14 and I-1-16.

Definition I-1-15: The polynomial P(t) related to a cyclic group code C as described in Theorem I-1-9 is said to be the generating polynomial of C.

Corollary: If P(t) is the generating polynomial of a cyclic group code C of length n, then P(t) divides $t^n - 1$.

Proof: Let Q(t) be a non-zero polynomial of highest degree such that Q(t) divides both P(t) and $t^n - 1$. Then there exist R(t) and S(t) such that $R(t)P(t) + S(t)(t^n - 1) = Q(t)$. Then Q(t) is in C* and is therefore actually a multiple of P(t); therefore P(t) divides $t^n - 1$.

Theorem I-1-10: There exists a one-to-one correspondence between cyclic group codes C of length n and monic polynomials P(t) which divide $t^n - 1$.

Proof: The correspondence from codes to polynomials is that established by Theorem I-1-9 and its corollary.

Conversely, given a monic polynomial P(t) which divides $t^n - 1$, we form the set C* of all multiples $P_x(t) = R(t)P(t)$ of P(t) of degree $\leq n - 1$. The set C* is clearly closed under the operations of component-wise addition, scalar multiplication, and multiplication by t mod $t^n - 1$. Therefore, the set of n-tuples x in C defined by the correspondence x in C $\longleftrightarrow$ $P_x(t)$ in C* is a cyclic group code with generating polynomial P(t).

Corollary 1: If C is any given cyclic group code and P(t) is its generating polynomial, then C is isomorphic to the set of all multiples of P(t) of degree $\leq n - 1$ under the correspondence x in C $\longleftrightarrow$ $P_x(t) = R(t)P(t)$ and with respect to the operations of componentwise addition, scalar multiplication, and multiplication by t mod $t^n - 1$.

Corollary 2: If C is a cyclic group code of length n with generating polynomial P(t) of degree m, then $N(C) = q^{n-m}$.

Theorem I-1-11: If C is a cyclic group code with generating polynomial P(t) and A is a matrix determining C, then for any two n-vectors u and v, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ iff $Au = Av$.

Proof: First suppose that $Au = Av$, and thus that $u - v$ is in C. Then $P_u(t) - P_v(t) = P_{u-v}(t) = R(t)P(t)$ for some R(t); therefore $P_u(t) \equiv P_v(t) \pmod{P(t)}$.

Conversely, suppose $P_u(t) \equiv P_v(t) \pmod{P(t)}$, which implies that $P_{u-v}(t) = P_u(t) - P_v(t) = R(t)P(t)$ for some R(t). Then $P_{u-v}(t)$ is in C*, $u - v$ is in C, and $A(u - v) = 0$; therefore $Au = Av$.

Theorem I-1-12: If C is a cyclic group code with generating polynomial P(t) and A is a matrix determining C, then C corrects a set of errors D iff $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies that u = v whenever u and v are in D.

Proof: First, suppose that C corrects D. Then by Theorem I-1-5, Au = Av implies that u = v whenever u and v are in D. Therefore, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies Au = Av implies u = v whenever u and v are in D.

Conversely, suppose that $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies that u = v whenever u and v are in D. By Theorem I-1-11, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ iff Au = Av. Therefore, Au = Av implies that u = v whenever u and v are in D, so C corrects D.

Theorem I-1-13: If A is any matrix of the form $A = \left[a, La, \ldots, L^{n-1}a\right]$ and x is any n-component column vector, then $Ax = P_x(L)a$.

Proof: $Ax = ax_1 + Lax_1 + \ldots + L^{n-2}ax_{n-2} + L^{n-1}ax_{n-1}$

$= x_1a + x_1La + \ldots + x_{n-1}L^{n-1}a$ because each x_i is a scalar

$= (x_1 + x_1L + \ldots + x_{n-1}L^{n-1})a$ because matrix multiplication is distributive over addition.

$= P_x(L)a$.

Theorem I-1-14: If C is a cyclic group code with generating polynomial P(t), then a suitable matrix A which determines C is one of the form $A = \left[a, La, \ldots, L^{n-1}a\right]$, where a is the m-component column vector $\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$ and L is the companion matrix $\begin{bmatrix} h_{m-1} & 1 & 0 & \ldots & 0 \\ h_{m-2} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ h_1 & 0 & 0 & \ldots & 1 \\ h_0 & 0 & 0 & \ldots & 0 \end{bmatrix}$

of the polynomial $P(t) = t^m + \sum_{i=0}^{m-1} h_i t^i$.

Proof: First, suppose that we are given a code C with generating polynomial P(t), and let us form the matrices L and A described above. Then P(t) is both the characteristic polynomial and the minimum polynomial of L. It is also the minimum polynomial satisfying $P(L)a = 0$, since the vectors $L^i a$ are all linearly independent for the a specified above and $i = 0, 1, \ldots, n-1$. This means that for every x in C, we have $Ax = 0$ since

$$Ax = ax_0 + Lax_1 + \ldots + L^{n-2}ax_{n-2} + L^{n-1}ax_{n-1}$$

$$= P_x(L)a$$

$$= R(L)\,P(L)a = 0.$$

Conversely, every x such that $Ax = 0$ is in C since $Ax = 0$ implies $ax_0 + Lax_1 + \ldots + L^{n-1}ax_{n-1} = 0$, which implies that $P_x(L)a = 0$. This implies that $P_x(t) = R(t)P(t)$ for some R(t) because P(t) is the minimum polynomial such that $P(L)a = 0$ and hence divides $P_x(t)$. By the Corollary 1 of Theorem I-1-10, the set of all multiples of degree $\leq n-1$ of the generating polynomial of a cyclic group code is isomorphic to the code itself. Therefore, x is in C.

We have proved that $Ax = 0$ iff x is in C; therefore, A determines C.

Definition I-1-16: When L and a have the form described in Theorem I-1-14 the matrix A is said to be the standard form corresponding to P(t).

Corollary: If A is an m x n matrix of rank m in standard form which determines a cyclic group code C with generating polynomial P(t) of degree m*, then $m^* = m$.

Proof: By the construction of A in Theorem I-1-14.

Theorem I-1-15: If A is an m x n matrix of rank m in standard form which determines a cyclic group code C with generating polynomial P(t), then for any n-tuple w we have $P_{Aw}(t) = \text{Res } P_w(t) (\text{mod } P(t))$.

Proof: Let $P_z(t) = \text{Res } P_w(t) (\text{mod } P(t))$. Then clearly $P_z(t)$ is of degree $\leq m - 1$. Let z* be the n-tuple $z^* = [\overbrace{0, \ldots, 0}^{n-m}, \overbrace{z_{m-1}, \ldots, z_0}^{m}]$. Then $Az^* = z$ since A is a matrix of the form $[B, I_m]$, where B is an m x (n - m) matrix and I_m is the identity matrix of order m. By hypothesis, $P_w(t) = R(t)P(t) + P_z(t) = P_x(t) + P_z(t)$ with x in C. Therefore, $w = (x + z^*)$, and $Aw = A(x + z^*) = Az^* = z$. Therefore, $P_{Aw}(t) = P_z(t) = \text{Res } P_w(t) (\text{mod } P(t))$.

Theorem I-1-16: A matrix A of rank m determines a cyclic group code C of length n with generating polynomial P(t) iff A is of the form $A = [a, La, \ldots, L^{n-2}a, L^{n-1}a]$, where L is an m x m matrix and a is a column vector such that $L^n a = a$ and such that P(t) is the minimum polynomial satisfying $P(L)a = 0$.

Proof: First, suppose that A is a matrix having the specified form and that x is any n-tuple; then $Ax = 0$ iff $P_x(L)a = 0$ iff $\text{Res } P_x(t) = 0 (\text{mod } P(t))$ iff $P_x(t) = R(t)P(t)$ for some R(t) such that (degree of R(t)) + (degree of P(t)) $\leq n - 1$. By Theorem I-1-4, there exists a group code C such that $Ax = 0$ iff x is in C. Combining these two results, we see that x is in C iff $P_x(t) = R(t)P(t)$ for some R(t) with (degree of R(t)) + (degree of P(t)) $\leq n - 1$; therefore, C is actually a cyclic group code with generating polynomial P(t).

Conversely, suppose that C is a cyclic group code with generating polynomial P(t). By Theorem I-1-14, we know that this is possible iff there exists a matrix B of the form $B = [b, Mb, \ldots, M^{n-2}b, M^{n-1}b]$, where M is the m x m companion matrix

$$\begin{bmatrix} h_{m-1} & 1 & 0 & \cdots & 0 \\ h_{m-2} & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ h_1 & 0 & 0 & \cdots & 1 \\ h_0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

of P(t) and b is $\begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$.

Since B and the matrix A determining C are both m x n matrices of rank m, the rows of B form a basis for the space of the rows of A; that is, there exists a non-singular matrix T such that $A = TB$. This means that A is a matrix of the form $A = \left[Tb, TMb, \ldots, TM^{n-2}b, TM^{n-1}b\right]$. Applying this same linear transformation T to b and M, we obtain a vector $a = Tb$ and a matrix $L = TMT^{-1}$ such that $TM^{j}b = (TM^{j}T^{-1})Tb = L^{j}a$. Therefore, A is of the form $A = \left[a, La, \ldots, L^{n-2}a, L^{n-1}a\right]$, where L is an m x m matrix, a is an m-vector, and $L^{n}a = (TM^{n}T^{-1})Tb = TM^{n}b = Tb = a$. Also, P(t) is the minimum polynomial satisfying $P(L)a = 0$ because $P(L)a = TP(M)b$.

Corollary: If A is any matrix of rank m determining a cyclic group code C with generating polynomial P(t) of degree m*, then $m^* = m$.

Section I-2. Binary Code-Generating Polynomials.

Definition I-2-1: An n-tuple x is said to be binary iff its coordinates $x_0, x_1, \ldots, x_{n-1}$ are elements of the field GF(2).

For convenience, from now on we shall consider only binary errors and code words and, because -1 = +1 in GF(2), we shall no longer use "-1" in such expressions as "mod $t^n - 1$" but instead will write "mod $t^n + 1$".

Definition I-2-2: The polynomial P(t) is said to be of period n iff P(t) divides $t^n + 1$ but divides no polynomial of the form $t^j + 1$, where $0 < j \leq n - 1$.

Definition I-2-3: The polynomial H(t) is said to be of order d with respect to P(t) iff d is the smallest positive integer such that $H(t) \equiv t^d H(t) \pmod{P(t)}$. We also say in this case that H(t) lies in a cycle of length d with respect to P(t). Note that because we always have $H(t) \equiv t^n H(t) \pmod{P(t)}$, where n is the period of P(t), we know that d must be a divisor of n.

Theorem I-2-1: The polynomial H(t) is of order d with respect to P(t) iff the polynomial $(P(t)/\gcd(H(t),P(t)))$ is of period d.

Proof: Let $P(t) = F(t)G(t)$, where $F(t) = \gcd(H(t),P(t))$ and $G(t) = (P(t)/\gcd(H(t),P(t)))$. Then H(t) is of order d with respect to P(t) iff d is the smallest positive integer such that $(t^d + 1)H(t) \equiv 0 \pmod{F(t)G(t)}$, which in turn is true iff d is the smallest positive integer such that $(t^d + 1)H(t)/F(t) \equiv 0 \pmod{G(t)}$. Since H(t)/F(t) is prime to G(t), we see that this can be true iff d is the smallest positive integer such that G(t) divides $t^d + 1$. Therefore, H(t) is of order d with respect to P(t) iff G(t) is of period d.

Corollary: If P(t) is irreducible and of period n, and P(t) does not divide H(t), then H(t) is of order n with respect to P(t).

Proof: If P(t) is irreducible and P(t) does not divide H(t), then gcd (H(t),P(t)) = 1; therefore, the order of H(t) with respect to P(t) is equal to the period of P(t)/1, which is n.

Definition I-2-4: The n-tuple u is said to be of order d iff $P_u(t)$ is of order d with respect to $t^n + 1$. We then have $u = u^{(d)}$, where $u^{(j)}$ is defined recursively by the formulas $u^{(0)} = u$, $u^{(1)} = u'$, and $u^{(j)} = u^{(j-1)'}$. Note that because we have $u = u^{(n)}$, we know that d must be a divisor of n.

Definition I-2-5: A set of cyclic errors D is said to be of order d iff every non-zero error in D is of order d.

Definition I-2-6: Let the elements of GF(2) be put into one-to-one correspondence with the integers 0 and 1. We say that v is the normal form and j is the index of the cyclic error u of order d iff $v^{(j)} = u$, where $0 \leq j \leq d - 1$ and $\sum_{i=0}^{n-1} v_i 2^i$ is a minimum for all cyclic shifts of v. Note that under these restrictions, v and j are unique for a given u.

Definition I-2-7: The polynomial $P_v(t)$ is said to be the normal polynomial of u, and any polynomial $P_v(t)$ for which $\sum_{i=0}^{n-1} v_i 2^i$ is a minimum for all cyclic shifts of v is a normal polynomial.

The following theorems indicate that each error u in a set of cyclic errors D is related through the operation of cyclic shifting, or multiplication by t (mod $t^n + 1$), to a unique normal form v, which may therefore be used to identify the cycle to which u belongs.

Theorem I-2-2: If D is a set of cyclic errors of length n and D* is the set of all polynomials $P_u(t)$ such that u is in D, then each element $P_u(t)$ in D* may be written uniquely in the form $P_u(t) = \text{Res } t^i H(t)$ (mod $t^n + 1$), where, if d is the order of u, H(t) is a normal polynomial

in D* of order d with respect to $t^n + 1$ and $o \leq i \leq d - 1$.

Proof: Suppose that u is in D and that $u = v^{(j)}$, where v is the normal form and j the index of u. Then $P_u(t) = P_{v^{(j)}}(t) = \text{Res } t^j P_v(t) \pmod{t^n + 1}$, where $o \leq j \leq d - 1$ and $P_v(t)$ is a normal polynomial in D* of order d with respect to $t^n + 1$. Taking $i = j$ and $H(t) = P_v(t)$, we obtain $P_u(t) = \text{Res } t^i H(t) \pmod{t^n + 1}$.

Theorem I-2-3: If D is a set of cyclic errors of length n and D* is the set of all polynomials $P_u(t)$ such that u is in D, then $\text{Res } t^i H(t) \pmod{t^n + 1}$ is in D* for any integer i and any polynomial H(t) in D*.

Proof: Suppose that i is any integer and that H(t) is any polynomial in D*. Then there exists an integer s and a polynomial M(t) such that $H(t) = \text{Res } t^s M(t) \pmod{t^n + 1}$, where $o \leq s \leq n - 1$ and M(t) is a normal polynomial in D*. Now let $P_v(t) = M(t)$ and let $j \equiv i + s \pmod{n}$. Then there exists a u in D such that $\text{Res } t^i H(t) \pmod{t^n + 1} = \text{Res } t^{i+s} M(t) \pmod{t^n + 1} = \text{Res } t^j P_v(t) \pmod{t^n + 1} = P_{v^{(j)}}(t) = P_u(t)$ in D*.

Theorem I-2-4: The cyclic group code C of length n with generating polynomial P(t) corrects a set of cyclic errors D iff, for any two normal polynomials $H_1(t)$ and $H_2(t)$ in D* both of order d with respect to $t^n + 1$, $t^{i_1} H_1(t) \equiv t^{i_2} H_2(t) \pmod{P(t)}$ implies $H_1(t) = H_2(t)$ and $i_1 \equiv i_2 \pmod{d}$.

Proof: If either $H_1(t)$ or $H_2(t)$ is equal to zero, then $t^{i_1} H_1(t) \not\equiv t^{i_2} H_2(t)$ unless $H_1(t) = H_2(t) = 0$; therefore, this is merely an application of Theorem I-1-12 with $P_u(t)$ replaced by $t^{i_1} H_1(t)$ and $P_v(t)$ replaced by $t^{i_2} H_2(t)$.

Theorem I-2-5: Let P(t) be a polynomial of degree m which generates the code C of length n correcting a set of cyclic errors D. Then for each u in D there exists a unique polynomial $K_u(t)$ of degree $\leq m - 1$ such that $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$.

Proof: Let A be the m x n matrix of rank m in standard form corresponding to P(t). Then $K_u(t) = P_{Au}(t) = \text{Res } P_u(t) \pmod{P(t)}$ is the desired polynomial.

Definition I-2-8: The polynomial $K_u(t)$ is said to be the image polynomial of u with respect to P(t).

Definition I-2-9: Let the elements of GF(2) be put into one-to-one correspondence with the integers 0 and 1 as in Definition I-2-6, let u be an error of order d in a set of cyclic errors D, and let $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$, as in Theorem I-2-5. Then we say that $R_u(t)$ is the representative polynomial of u with respect to P(t) and j is the power of u with respect to P(t) iff $K_u(t) = \text{Res } t^j R_u(t) \pmod{P(t)}$, where $0 \leq j \leq d - 1$, and, for all polynomials R(t) such that $R_u(t) = \text{Res } t^h R(t) \pmod{P(t)}$, we have $\sum_{i=0}^{m-1} r_i 2^i \geq \sum_{i=0}^{m-1} (r_u)_i 2^i$, where h is any integer and the r_i's and $(r_u)_i$'s are the coefficients of R(t) and $R_u(t)$ respectively.

Definition I-2-10: The polynomial R(t) of degree $\leq m - 1$ is said to be a representative polynomial with respect to P(t) iff it has the property that for all polynomials K(t) such that $K(t) \equiv t^j R(t) \pmod{P(t)}$, we have $\sum_{i=0}^{m-1} k_i 2^i \geq \sum_{i=0}^{m-1} r_i 2^i$, where the k_i's and r_i's are the coefficients of K(t) and R(t) respectively.

Theorem I-2-6: If P(t) is the generating polynomial of a code which corrects the set of cyclic errors D of length n and D# is the set of all polynomials $K_u(t)$ such that $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$, where u is in D, u is of order d, and $K_u(t)$ is of order d with respect to P(t), then each

element of D# may be written uniquely in the form $K_u(t) = \text{Res } t^i H(t) \pmod{P(t)}$, where $0 \leq i \leq d - 1$ and $H(t)$ is a representative polynomial in D# of order d with respect to $P(t)$.

Proof: Suppose that u in D is of order d and that $K_u(t)$ in D# is also of order d with respect to $P(t)$ and is the image polynomial of u with respect to $P(t)$. Then $K_u(t) = \text{Res } t^j R(t) \pmod{P(t)}$, where j is the power of u with respect to $P(t)$, $0 \leq j \leq d - 1$, and $R(t)$ is the representative polynomial of u with respect to $P(t)$. Taking $i = j$ and $H(t) = R(t)$, we obtain $K_u(t) = \text{Res } t^i H(t) \pmod{P(t)}$.

Theorem I-2-7: If $P(t)$ is the generating polynomial of a code which corrects the set of cyclic errors D of length n and D# is the set of all polynomials $K_u(t)$ such that $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$, where u is in D, then $\text{Res } t^i H(t) \pmod{P(t)}$ is in D# for any integer i and any polynomial $H(t)$ in D#.

Proof: Suppose that i is any integer and $H(t)$ is any polynomial in D#. Then there exists an integer s and a polynomial $M(t)$ such that $H(t) = \text{Res } t^s M(t) \pmod{P(t)}$, where $0 \leq s \leq d - 1$ and $M(t)$ is a representative polynomial with respect to $P(t)$ and is in D#. Now let $R(t) = M(t)$ and let $j \equiv i + s \pmod{n}$. Then there exists a u in D such that $\text{Res } t^i H(t) \pmod{P(t)} = \text{Res } t^{i+s} M(t) \pmod{P(t)} = \text{Res } t^j R(t) \pmod{P(t)} = K_u(t)$ in D#.

Theorem I-2-8: The cyclic group code C of length n with generating polynomial $P(t)$ corrects the set of cyclic errors D iff, for any two representative polynomials $H_1(t)$ and $H_2(t)$ in D# both of order d with respect to $t^n + 1$, $t^{i_1} H_1(t) \equiv t^{i_2} H_2(t) \pmod{P(t)}$ implies $H_1(t) = H_2(t)$ and $i_1 \equiv i_2 \pmod{d}$.

Proof: If either $H_1(t)$ or $H_2(t)$ is equal to zero, then $t^{i_1}H_1(t) \not\equiv t^{i_2}H_2(t) \pmod{P(t)}$ unless $H_1(t) = H_2(t) = 0$; therefore, this is merely an application of Theorem I-1-12 with $P_u(t)$ replaced by $t^{i_1}H_1(t)$ and $P_v(t)$ replaced by $t^{i_2}H_2(t)$.

Theorem I-2-9: The cyclic group code C of length n with generating polynomial P(t) of degree m corrects the set of cyclic errors D iff for any u and v in D, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies $P_u(t) \equiv P_v(t) \pmod{t^n + 1}$.

Proof: Since $P_u(t) \equiv P_v(t) \pmod{t^n + 1}$ iff $u = v$, this result follows directly from Theorem I-1-12.

Theorem I-2-10: The cyclic group code C of length n with generating polynomial P(t) corrects the set of cyclic errors D iff for each distinct normal form v of order d in D there exists a distinct representative polynomial $R_v(t)$ in D# of order d with respect to P(t).

Proof: Suppose first that P(t) generates C and that u and v are distinct normal forms of order d in D, but that $R_u(t) = R_v(t)$. Then we know that there exists a j, $0 \leq j \leq d - 1$, such that $K_u(t) \equiv t^j K_v(t) \pmod{P(t)}$. This means that $P_u(t) \equiv t^j P_v(t) \pmod{P(t)}$, which means (because C corrects D) in turn that $P_u(t) \equiv t^j P_v(t) \pmod{t^n + 1}$, and, finally that $u = v^{(j)}$. Also, if u is any error of order d in D, then $R_u(t)$ is of order d with respect to P(t) because $R_u(t) \equiv t^j R_u(t) \pmod{P(t)}$ implies, as before, that $u = u^{(j)}$, which is impossible unless $j \equiv 0 \pmod{d}$.

Conversely, suppose that for each distinct normal form w of order d in D there exists a distinct representative polynomial $R_w(t)$ in D# of order d with respect to P(t) and suppose that $P_u(t) \equiv P_v(t) \pmod{P(t)}$, where u and v are any errors of order d in D. Then $R_u(t) = R_v(t)$, which by hypothe-

sis implies that u and v have the same normal form, so that $u = v^{(j)}$, for some $0 \leq j \leq d - 1$. Since, however, $P_u(t) \equiv t^0 P_v(t) \pmod{P(t)}$ and $P_u(t)$ and $P_v(t)$ are of order d with respect to P(t), we see that $j = 0$. Therefore, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies that $u = v$.

Theorem I-2-11: The cyclic group code C of length n with generating polynomial P(t) corrects the set of cyclic errors D iff $t^n + 1$ corrects the set of errors $\hat{D}$ which has the property that $\hat{u}$ is in $\hat{D}$ iff there exists a u in D such that $P_{\hat{u}}(t) \equiv P_u(t)Q(t) \pmod{t^n + 1}$, where $Q(t) = (t^n + 1)/P(t)$.

Proof: We know that $P_u(t)Q(t) \equiv P_v(t)Q(t) \pmod{P(t)Q(t)}$ iff $P_u(t) \equiv P_v(t) \pmod{P(t)}$, which means that P(t) generates a code correcting D iff $P(t)Q(t) = t^n + 1$ generates a code correcting $\hat{D}$.

The preceding theorems indicate that 1) if we know which polynomials of degree $< m$ lie in which cycles of P(t), then we can readily determine whether or not the code generated by P(t) corrects any given set of cyclic errors, and 2) each cycle of P(t) possesses a unique representative polynomial and length. Therefore, we can test whether or not a given set of cyclic errors is corrected by the code generated by P(t) in the following way: we first find the set of normal forms of the set of cyclic errors and the order of each of these. This can usually be done quite easily by inspection. Then, for each normal form, we find first the corresponding image polynomial with respect to P(t) and then the corresponding representative polynomial with respect to P(t) and its order with respect to P(t). If each distinct normal form corresponds to a distinct representative polynomial whose order with respect to P(t) is the same as the order of the normal form, then Theorem I-2-10 tells us that the code generated by P(t)

corrects D. Alternatively, Theorem I-2-11 shows that each error u in the set of cyclic errors D may be first multiplied modulo $t^n + 1$ by $Q(t) = (t^n + 1)/P(t)$ to form a modified error $\hat{u}$; if each distinct normal form v in the original set of errors D corresponds to a distinct normal form $\hat{v}$ from the set of modified errors $\hat{D}$, and each $\hat{v}$ is of the same order as its corresponding v, then the code generated by P(t) corrects D. The method of Theorem I-2-10 is best adapted to use on a high-speed computer, whereas the method of Theorem I-2-11 can be carried out by hand even for fairly long and complicated codes because the normal forms can be obtained by inspection.

If we are not interested in determining whether the code generated by a given polynomial P(t) corrects a given set of errors D but rather in examining the actual cycle structure of P(t), we may proceed as follows: First, we note that the polynomial 0 lies by itself in a cycle of length 1. Next, we choose the polynomial 1 as the representative polynomial of the first non-zero cycle of P(t), then form and write down the products $t, t^2, \ldots, t^{m-1}, t^m \pmod{P(t)}, \ldots, t^{n-1} \pmod{P(t)}$, where $t^n \equiv 1 \pmod{P(t)}$. We know that $n \leq 2^m - 1$ because the total number of non-zero binary polynomials of degree $< m$ is $2^m - 1$. If $n = 2^m - 1$, then the cycle structure of P(t) has been completely determined -- namely, it consists of a zero cycle of length 1 and a non-zero cycle of length $2^m - 1$. If $n < 2^m - 1$, we may obtain the remaining cycles of P(t) by choosing that polynomial H(t) of degree $< m$ not appearing in any cycle already obtained for which $\sum_{i=0}^{m-1} h_i 2^i$ is a minimum as the representative polynomial of the next non-zero cycle and writing down the products Res $tH(t) \pmod{P(t)}$, $\ldots$, Res $t^{d-1}H(t) \pmod{P(t)}$, where d is the order of H(t) with respect to P(t). This procedure is repeated until all polynomials of degree $< m$ have been accounted for.

Section I-3. Factorable Code-Generating Polynomials.

Let us now suppose that $P(t)$ is a polynomial of the form $P(t) = P_2(t)P_1(t)$ and let us consider whether or not it is possible to relate the error-correcting properties of $P(t)$ to the cycle structures of its factors $P_1(t)$ and $P_2(t)$.

Theorem I-3-1: If $P(t) = P_2(t)P_1(t)$, where $P_2(t)$ and $P_1(t)$ are relatively prime, and each of $P(t)$, $P_1(t)$, and $P_2(t)$ is of period n, n_1, and n_2 respectively, then $n = \text{lcm}(n_1, n_2)$.

Proof: If $P(t) = P_2(t)P_1(t)$, where $P_2(t)$ and $P_1(t)$ are relatively prime, then $P(t)$ divides $Q(t)$ iff $P_2(t)$ divides $Q(t)$ and $P_1(t)$ divides $Q(t)$.

Theorem I-3-2: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime. Let $H(t)$ and $K(t)$ be any two polynomials such that $H(t)$ and $K(t)$ are both of order d_1 with respect to $P_1(t)$ and both of order d_2 with respect to $P_2(t)$. Then 1) $H(t)$ and $K(t)$ are both of order $d = \text{lcm}(d_1, d_2)$ with respect to $P(t)$ and 2) there exists a j such that $H(t) \equiv t^j K(t) \pmod{P(t)}$ iff there exist two integers j_1 and j_2 such that $0 \leq j_1 \leq d_1 - 1$, $0 \leq j_2 \leq d_2 - 1$, $H(t) \equiv t^{j_1} K(t) \pmod{P_1(t)}$, $H(t) \equiv t^{j_2} K(t) \pmod{P_2(t)}$, and $j_1 \equiv j_2 \pmod{\gcd(d_1, d_2)}$.

Proof: 1) Since $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime, $H(t) \equiv t^d H(t) \pmod{P(t)}$ implies that $H(t) \equiv t^d H(t) \pmod{P_1(t)}$. However, d_1 is by hypothesis the smallest positive integer such that $H(t) \equiv t^{d_1} H(t) \pmod{P_1(t)}$; therefore, we see that d_1 divides d. Similarly, d_2 divides d, and therefore $H(t) \equiv t^d H(t) \pmod{P(t)}$ for any d which is a multiple of both d_1 and d_2. The smallest positive integer which is a multiple of both d_1 and d_2 is $\text{lcm}(d_1, d_2)$, therefore the order of $H(t)$ with respect to $P(t)$ is $d = \text{lcm}(d_1, d_2)$. Similarly, the order of $K(t)$

with respect to $P(t)$ is $d = \text{lcm}(d_1, d_2)$.

2) Since $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime, we have $H(t) \equiv t^jK(t) \pmod{P(t)}$ iff $H(t) \equiv t^jK(t) \pmod{P_1(t)}$ and $H(t) \equiv t^jK(t) \pmod{P_2(t)}$. Because $H(t)$ and $K(t)$ are of order d_1 with respect to $P_1(t)$, $H(t) \equiv t^jK(t) \pmod{P_1(t)}$ iff there exist integers c_1 and j_1, with $0 \leq j_1 \leq d_1 - 1$, such that $j = c_1d_1 + j_1$ and $H(t) \equiv t^{j_1}K(t) \pmod{P_1(t)}$. Similarly, $H(t) \equiv t^jK(t) \pmod{P(t)}$ iff there exist integers c_2 and j_2 with $0 \leq j_2 \leq d_2 - 1$, such that $j = c_2d_2 + j_2$ and $H(t) \equiv t^{j_2}K(t) \pmod{P_2(t)}$. Finally, $c_1d_1 + j_1 = j$ and $c_2d_2 + j_2 = j$ iff $j_1 - j_2 = c_1d_1 + c_2d_2$, which in turn is true iff $j_1 - j_2 \equiv 0 \pmod{\gcd(d_1, d_2)}$.

Theorem I-3-3: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime. Then the cyclic group code C of length n with generating polynomial $P(t)$ corrects the set of cyclic errors D iff, for each pair of distinct normal forms u and v of order d in D, we have: 1) $P_u(t)$ and $P_v(t)$ are both of order d with respect to $P(t)$ and 2) it is not true that there exist two integers j_1 and j_2 such that $0 \leq j_1 \leq d_1 - 1$, $0 \leq j_2 \leq d_2 - 1$, $P_u(t) \equiv t^{j_1}P_v(t) \pmod{P_1(t)}$, $P_u(t) \equiv t^{j_2}P_v(t) \pmod{P_2(t)}$, and $j_1 \equiv j_2 \pmod{\gcd(d_1, d_2)}$, where d_1 is the order of $P_u(t)$ and $P_v(t)$ with respect to $P_1(t)$ and d_2 is the order of $P_u(t)$ and $P_v(t)$ with respect to $P_2(t)$.

Proof: By Theorem I-2-4, C corrects D iff 1) $P_u(t)$ and $P_v(t)$ are both of order d with respect to $P(t)$ and 2) there exists no j such that $P_u(t) \equiv t^jP_v(t) \pmod{P(t)}$ whenever u and v are distinct normal forms in D. Therefore, this is a direct application of Theorems I-2-4 and I-3-2.

Theorems I-3-2 and I-3-3 show that we can indeed predict the error-correcting properties of the product $P(t)$ if we know the cycle structures

of the factors $P_1(t)$ and $P_2(t)$. Thus, to determine whether or not a given set of cyclic errors D is corrected by the code generated by $P(t)$, we first find the set of normal forms in D. Let u be such a normal form and let $K_u(t)$, $K_{1u}(t)$, and $K_{2u}(t)$ be its image polynomials with respect to $P(t)$, $P_1(t)$, and $P_2(t)$ respectively; and let v be another normal form with image polynomials $K_v(t)$, $K_{1v}(t)$, and $K_{2v}(t)$. By Theorem I-3-3, we know that there exists a j such that $K_u(t) \equiv t^j K_v(t) \pmod{P(t)}$ iff there exist two integers j_1 and j_2 such that $0 \leq j_1 \leq d_1 - 1$, $0 \leq j_2 \leq d_2 - 1$, $K_{1u}(t) \equiv t^{j_1} K_{1v}(t) \pmod{P_1(t)}$, $K_{2u}(t) \equiv t^{j_2} K_2(t) \pmod{P_2(t)}$, and $j_1 \equiv j_2 \pmod{\gcd(d_1, d_2)}$. Assuming that the cycle structures of $P_1(t)$ and $P_2(t)$ have been found and written down as described at the end of Section I-2, it is easy to determine whether or not a j_1 and j_2 exist satisfying these conditions.

The following theorems are important because they provide the theoretical basis for the method used in succeeding chapters to construct an arbitrarily long code correcting a given set of errors if we already know of a shorter code which corrects a set of errors with a related set of normal forms.

Theorem I-3-4: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that $P(t)$ is of period $n = \text{lcm}(n_1, n_2)$. Let D be a set of cyclic errors of length n and let each non-zero polynomial $P_u(t)$ in D* be of order n_1 with respect to $P_1(t)$. Finally, suppose that for each distinct non-zero normal form v in D here exists a distinct polynomial $R_{2v}(t)$ which is the representative polynomial of $P_v(t)$ with respect to $P_2(t)$ and is of order n_2 with respect to $P_2(t)$. Then D is of order n and the code C generated by $P(t)$ corrects D.

Proof: Let u and v be any two distinct non-zero normal forms in D with representative polynomials $R_u(t)$ and $R_v(t)$ with respect to $P(t)$, and suppose that $R_u(t) = R_v(t)$. Then there exists a j such that $K_u(t) \equiv t^j K_v(t)$ (mod $P(t)$), which means that $K_u(t) \equiv t^j K_v(t)$ (mod $P_2(t)$), which contradicts the hypothesis that each distinct non-zero normal form v has a distinct representative polynomial with respect to $P_2(t)$. Also, Theorem I-3-2 shows that each representative polynomial $R_u(t)$ in D# is of order $n = \text{lcm}\ (n_1, n_2)$ with respect to $P(t)$ because $P_u(t)$ is of order n_1 with respect to $P_1(t)$ and of order n_2 with respect to $P_2(t)$. Therefore, D is of order n and, by Theorem I-2-10, C corrects D.

Corollary: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that $P(t)$ is of period $n = \text{lcm}(n_1, n_2)$. Let D be a set of cyclic errors of length n and suppose that $P_1(t)$ is irreducible and divides no non-zero polynomial $P_u(t)$ in D*. Finally, suppose that for each distinct non-zero normal form v in D there exists a distinct polynomial $R_{2v}(t)$ which is the representative polynomial of $P_v(t)$ with respect to $P_2(t)$ and is of order n_2 with respect to $P_2(t)$. Then D is of order n and the code C generated by $P(t)$ corrects D.

Proof: By the Corollary of Theorem I-2-1, we know that if $P_1(t)$ is irreducible, is of period n_1, and does not divide $P_u(t)$, then $P_u(t)$ is of order n_1 with respect to $P_1(t)$.

Theorem I-3-5: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that $P(t)$ is of period $n = \text{lcm}(n_1, n_2)$. Let D_2

be a set of cyclic errors of order n_2 corrected by the code C_2 generated by $P_2(t)$, and let D be a set of cyclic errors whose set of normal forms is obtained from the set of errors D_2 as follows: to each non-zero normal polynomial $P_{u_2}(t)$ in D_2^* we first add some multiple $Q(t)P_2(t)$ of $P_2(t)$ of degree $\leq n - 1$, then form the representative polynomial $P_v(t)$ of $P_u(t) = P_{u_2}(t) + Q(t)P_2(t)$ with respect to $t^n + 1$, so that v is the normal form of the error u. Finally, suppose that each $P_v(t)$ so formed is of order n_1 with respect to $P_1(t)$. Then the code C generated by P(t) corrects D.

Proof: Because C_2 corrects D_2, we know by Theorem I-2-10 that for each distinct normal form in D_2 there exists a distinct representative polynomial of order n_2 with respect to $P_2(t)$. Because $P_u(t)$ differs from $P_{u_2}(t)$ only by a multiple of $P_2(t)$, we see that $P_u(t)$, and hence $P_v(t)$, has this same representative polynomial with respect to $P_2(t)$. Therefore, by Theorem I-3-4, C corrects D because each distinct non-zero normal form v has a distinct representative polynomial with respect to $P_2(t)$.

Corollary: Let P(t) be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that P(t) is of period $n = \text{lcm}(n_1, n_2)$. Let D_2 be a set of cyclic errors of order n_2 corrected by the code C_2 generated by $P_2(t)$, and let D be a set of cyclic errors whose set of normal forms is obtained from the set of errors D_2 as follows: to each non-zero normal polynomial $P_{u_2}(t)$ in D_2^* we first add some multiple $Q(t)P_2(t)$ of $P_2(t)$ of degree $\leq n - 1$, then form the representative polynomial $P_v(t)$ of $P_u(t) = P_{u_2}(t) + Q(t)P_2(t)$ with respect to $t^n + 1$, so that v is the normal form of the error u. Finally, suppose that $P_1(t)$ is irreducible and does not divide any error polynomial $P_u(t)$ in D^*. Then the code C generated by P(t) corrects D.

Proof: By the Corollary of Theorem I-2-1, we know that if $P_1(t)$ is irreducible, is of period n_1, and does not divide $P_u(t)$, then $P_u(t)$ is of order n_1 with respect to $P_1(t)$.

Section I-4. Efficiency.

Our principal concern in subsequent chapters will be with sets of cyclic errors which contain both the zero error and ng non-zero errors corresponding to a set of g distinct normal forms of order n. For such sets of errors, we now introduce the concept of "maximally efficient", which is intermediate between "efficient" (Definition I-1-12) and "close-packed" (Definition I-1-9).

Definition I-4-1: Let C be a cyclic group code of length n with generating polynomial P(t) of degree m which corrects a set of errors D containing the zero error and g distinct non-zero normal forms of order n so that $N(D) = ng + 1$. We say that C is maximally efficient with respect to D iff $ng + 1 \geq 2^m - g$.

Theorem I-4-1: Let C be a cyclic group code of the kind described in Definition I-4-1. If C is maximally efficient with respect to D, then there exists no cyclic group code $\hat{C}$ of length $\hat{n} \geq n + 1$ with generating polynomial $\hat{P}(t)$ of degree m which corrects D.

Proof: If a code $\hat{C}$ exists, then for such a code we have $\hat{n}g + 1 \geq (n + 1)g + 1 > 2^m$ because $ng + 1 > 2^m - g$. Therefore, for such a code, $N(C) \times N(D) = 2^{\hat{n} - m}(\hat{n}g + 1) > 2^{\hat{n}}$, which is impossible by Theorem I-1-1. Therefore, no such code $\hat{C}$ exists.

The following three theorems show how this new concept is related to the two earlier ones.

Theorem I-4-2: Let C be a cyclic group code of the kind described in Definition I-4-1. Then C is efficient iff $ng + 1 > 2^{m-1}$.

Proof: By Corollary 2 of Theorem I-1-10, we know that $N(C) = 2^{n-m}$, so that in this case $N(C) \times N(D) > 2^{n-m} \times 2^{m-1} = 2^{n-1}$. Therefore, by Definition I-1-12, C is efficient with respect to D.

Theorem I-4-3: Let C be a cyclic group code of the kind described in Definition I-4-1. Then C is maximally efficient iff $ng + 1 > 2^m - g$.

Proof: This is merely a restatement of Definition I-4-1.

Theorem I-4-4: Let C be a cyclic group code of the kind described in Definition I-4-1. Then C is close-packed iff $ng + 1 = 2^m$.

Proof: By Corollary 2 of Theorem I-1-10, we know that $N(C) = 2^{n-m}$, so that in this case $N(C) \times N(D) = 2^{n-m} \times 2^m = 2^n$. Therefore, by the Corollary of Theorem I-1-3, C is close-packed with respect to D.

Definition I-4-2: Let C be a cyclic group code of length n with generating polynomial P(t) of degree m which corrects the set of cyclic errors D. The efficiency of C with respect to D is $N(D)/2^m$.

Let C be a cyclic group code of the kind described in Definition I-4-1. It is immediately apparent that C is efficient with respect to D iff its efficiency with respect to D is $> \frac{1}{2}$; C is maximally efficient with respect to D iff its efficiency with respect to D is $> 1 - g/2^m$; and C is close-packed with respect to D iff its efficiency with respect to D is = 1.

Recalling Theorem I-1-1, we may summarize by saying that the code C is efficient with respect to D iff it is information-theoretically impossible to find a code $\hat{C}$ with generating polynomial $\hat{P}(t)$ of degree $\hat{m} < m$ which corrects a set of cyclic errors $\hat{D}$ as large as D. The code C is maximally

efficient with respect to D iff it is information-theoretically impossible to find a code $\hat{C}$ of length $\hat{n} > n$ with generating polynomial $\hat{P}(t)$ of degree $\hat{m} \leq m$ which corrects a set of cyclic errors $\hat{D}$ of order $\hat{n}$ with g distinct non-zero normal forms. The code C is close-packed with respect to D iff it is information-theoretically impossible to find a code $\hat{C}$ with generating polynomial $\hat{P}(t)$ of degree $\hat{m} \leq m$ which corrects a set of errors $\hat{D}$ containing a larger number of errors than D.

The concept of efficiency can be very useful when we are searching systematically for a polynomial which generates the most efficient possible code correcting a set of errors D such that the order of each non-zero error is equal to its length and such that D has a given set of g distinct non-zero normal forms. We first order the polynomials of each degree $m \geq 3$ (degrees 1 and 2 are trivial) with coefficients in GF(2) in such a way that P(t) precedes Q(t) iff either 1) degree of P(t) < degree of Q(t) or 2) degree of P(t) = degree of Q(t) = m and $\sum_{i=1}^{m} p_i 2^i < \sum_{i=1}^{m} q_i 2^i$. We next examine the cycle structure of the first polynomial P(t) for which $ng + 1 \leq 2^m$, where m is the degree of P(t) and n is its period, to ascertain whether or not the code C with generating polynomial P(t) corrects the set of cyclic errors D of length n and order n with the given set of g distinct non-zero normal forms. If C does correct D, we calculate the efficiency $(ng + 1)/2^m$ of C with respect to D before going on to the next P(t), so that we may now speed up our search by ignoring not only those P(t)'s for which $(ng + 1)/2^m > 1$, but also those for which $(ng + 1)/2^m$ is less than or equal to the efficiency of the code already found. Each time a new code is found, we note its efficiency and henceforth ignore any succeeding P(t) which does not give promise of generating a code whose efficiency is greater than that of the most efficient already found, continuing in this way until we find a code of efficiency 1 or stop for some other reason.

CHAPTER II

CODES DEFINED ON RASTERS

Section II-1. Correction of Cyclic Spot-Errors Defined on a Raster.

Definition II-1-1: A raster R is an ordered n-tuple $(r_0, r_1, \ldots, r_{n-1})$ of points in a space E such that for each pair (r_i, r_j) in R there is defined a distance function $m(r_i, r_j)$ with the following properties:

1) $m(r_i, r_i) = 0$ for each r_i in R

2) $m(r_i, r_j) > 0$ for all pairs $r_i \neq r_j$

3) $m(r_i, r_j) + m(r_j, r_k) \geq m(r_i, r_k)$ for all triples (r_i, r_j, r_k) in R

4) $m(r_i, r_j)$ depends only on the value of $i - j \pmod n$, so that there is a function M(s) defined over the integers with the property that $M(s) = M(q)$ if $s \equiv q \pmod n$ and $M(j - i) = m(r_i, r_j) = m(r_j, r_i) = M(i - j)$.

Examples of rasters which we shall consider are:

1) a straight line with raster points one unit apart, so that $m(r_j, r_{j+1}) = M(1) = 1$. Since we are considering only cyclic codes, we imagine that the last or $(n - 1)$ st point of the raster is at a distance of one unit from the first point, so that $m(r_{n-1}, r_n) = m(r_{n-1}, r_0) = M(1) = 1$. Note that because of its cyclic character, this raster may conveniently be regarded as a circle instead of as a straight line (see Figure II-1).

2) a two-dimensional square, rectangular, or hexagonal array with points one unit apart in a direction defined as "horizontal". If the raster is a b_1 by b_2 array, then points $r_0, r_1, \ldots, r_{b_1 - 1}$ are in the first row, $r_{(j-1)b_1}, r_{(j-1)b_1 + 1}, \ldots, r_{jb_1 - 1}$ are in the jth row, and $r_{(b_2 - 1)b_1}, r_{(b_2 - 1)b_1 + 1}, \ldots, r_{b_2 b_1 - 1}$ are in the last row. We again imagine that $m(r_{jb_1 - 1}, r_{jb_1}) = 1$ and $m(r_{b_2 b_1 - 1}, r_0) = 1$, and that the direction from $r_{jb_1 - 1}$ to r_{jb_1} is also

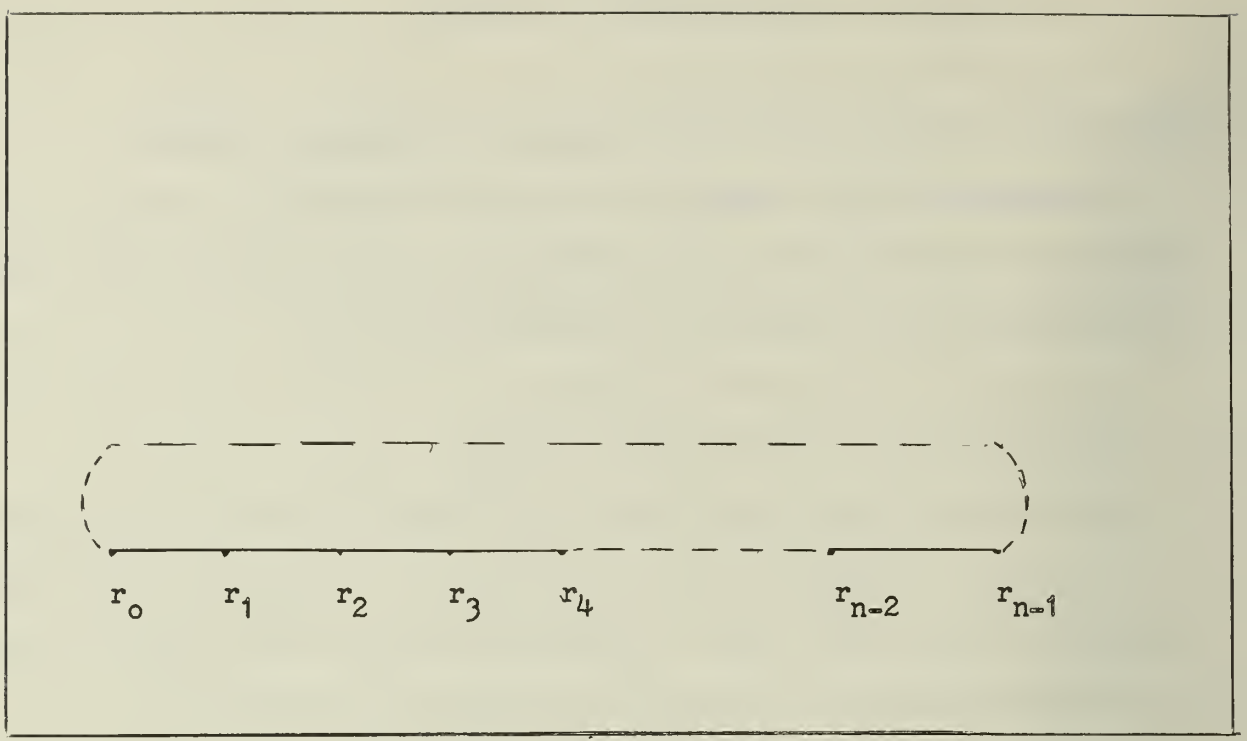


Figure II-1

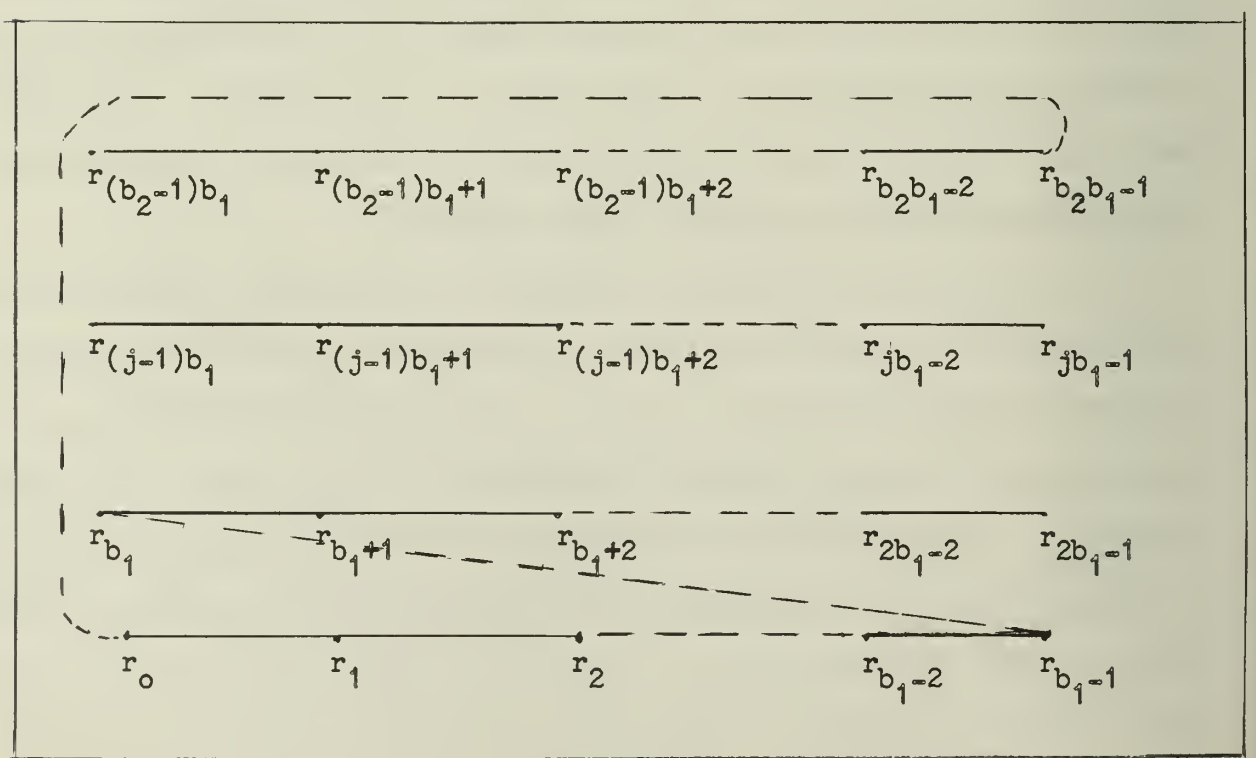


Figure II-2

"horizontal". Note that because of its cyclic character, this raster may conveniently be regarded as a helix instead of as a plane array (see Figure II-2).

Definition II-1-2: A spot is any subset of a raster R.

Definition II-1-3: The diameter of a spot S is $\max_{(r_i, r_j \text{ in } S)} m(r_i, r_j)$.

Note that the diameter of a spot S as defined here is the smallest value a such that S is wholly contained in each circle of radius a with center at one of the points of S, and therefore may be less than the diameter of the smallest physical circle enclosing all the points of S. For example, the vertices of an equilateral triangle with sides of length 1 form a spot of diameter 1 by Definition II-1-3, but the smallest circle enclosing all three points has diameter $2/\sqrt{3} = 1.1547\ldots$.

Definition II-1-4: An n-tuple $x = (x_0, x_1, \ldots, x_{n-1})$, where each coordinate is a symbol from GF(2), is said to be defined on a raster R iff the elements of R and x are in one-to-one correspondence with each other under the rule $r_i \longleftrightarrow x_i$; $i = 0, 1, \ldots, n-1$.

Definition II-1-5: A code C is said to be defined on a raster R iff each code word in C is defined on R, and similarly a set of errors D is said to be defined on R iff each error in D is defined on R.

Definition II-1-6: An element u in a set of cyclic errors D defined on a raster R is said to be a spot-error of diameter a iff the set of non-zero coordinates of u correspond to a spot of diameter a. Then we say that $m(u) = a$.

Definition II-1-7: A polynomial $P_x(t) = x_0 + x_1 t + \ldots + x_{n-1} t^{n-1}$ is said to be of diameter a iff x is a spot-error of diameter a.

Theorem II-1-1: If D is the set of all spot-errors of diameter $\leq$ a and D* is the set of all polynomials $P_u(t)$ such that u is in D and u is of order d, then each element $P_u(t)$ in D* may be written uniquely in the form $P_u(t) = \text{Res } t^i H(t) \pmod{t^n + 1}$, where H(t) is a normal polynomial of diameter $\leq$ a and $0 \leq i \leq d - 1$.

Proof: This is merely a special case of Theorem I-2-2, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-2: If D is the set of all spot-errors of diameter $\leq$ a and D* is the set of all polynomials $P_u(t)$ such that u is in D and u is of order d, then Res $t^i H(t) \pmod{t^n + 1}$ is in D* for any integer i and any polynomial H(t) of diameter $\leq$ a.

Proof: This is merely a special case of Theorem I-2-3, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Definition II-1-8: A cyclic group code C which corrects all spot-errors of diameter $\leq$ a is said to be a spot-correcting code of diameter a.

Theorem II-1-3: The cyclic group code C of length n with generating polynomial P(t) is a spot-correcting code of diameter a iff, for any two normal polynomials $H_1(t)$ and $H_2(t)$ of diameter $\leq$ a and both of order d with respect to $t^n + 1$, $t^{i_1} H_1(t) \equiv t^{i_1} H_2(t) \pmod{P(t)}$ implies $H_1(t) = H_2(t)$ and $i_1 \equiv i_2 \pmod{d}$.

Proof: This is merely a special case of Theorem I-2-4, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-4: If D is the set of all spot-errors of diameter $\leq$ a and D# is the set of all polynomials $K_u(t)$ such that $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$, where u is in D, u is of order d, and $K_u(t)$ is of order d with respect to P(t), then each element of D# may be written uniquely in the form $K_u(t) = \text{Res } t^i H(t) \pmod{P(t)}$, where $0 \leq i \leq d - 1$ and H(t) is a representative polynomial in D# of order d with respect to P(t).

Proof: This is merely a special case of Theorem I-2-6, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-5: If D is the set of all spot-errors of diameter $\leq$ a and D# is the set of all polynomials $K_u(t)$ such that $K_u(t) = \text{Res } P_u(t) \pmod{P(t)}$, where u is in D, then $\text{Res } t^i H(t) \pmod{P(t)}$ is in D# for any integer i and any polynomial H(t) in D#.

Proof: This is merely a special case of Theorem I-2-7, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-6: The cyclic group code C of length n with generating polynomial P(t) is a spot-correcting code of diameter a iff, for any two representative polynomials $H_1(t)$ and $H_2(t)$ of diameter $\leq$ a and both of order d with respect to $t^n + 1$, $t^{i_1} H_1(t) \equiv t^{i_2} H_2(t) \pmod{P(t)}$ implies $H_1(t) = H_2(t)$ and $i_1 \equiv i_2 \pmod{d}$.

Proof: This is merely a special case of Theorem I-2-8, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-7: The cyclic group code C of length n with generating polynomial P(t) is a spot-correcting code of diameter a iff for any u and v of diameter $\leq$ a, $P_u(t) \equiv P_v(t) \pmod{P(t)}$ implies $P_u(t) \equiv P_v(t) \pmod{t^n + 1}$.

Proof: This is merely a special case of Theorem I-2-9, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-8: The cyclic group code C of length n with generating polynomial P(t) is a spot-correcting code of diameter a iff for each distinct normal form v of diameter $\leq$ a and order d there exists a distinct representative polynomial $R_v(t)$ of order d with respect to P(t).

Proof: This is merely a special case of Theorem I-2-10, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-9: The cyclic group code C of length n with generating polynomial P(t) is a spot-correcting code of diameter a iff $t^n + 1$ corrects the set of errors $\hat{D}$ which has the property that $\hat{u}$ is in $\hat{D}$ iff there exists a u of diameter $\leq$ a such that $P_u(t) \equiv P_v(t)Q(t) \pmod{t^n + 1}$, where $Q(t) = (t^n + 1)/P(t)$.

Proof: This is merely a special case of Theorem I-2-11, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-10: Let P(t) be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that P(t) is of period $n = \text{lcm}(n_1, n_2)$. Then the cyclic group code C with generating polynomial P(t) is a spot-correcting code of diameter a iff for each pair of distinct normal forms u and v of diameter $\leq$ a and order d, we have: 1) $P_u(t)$ and $P_v(t)$ are both of order d with respect to P(t) and 2) it is not true that there exist two integers j_1 and j_2 such that $0 \leq j_1 \leq d_1 - 1$, $0 \leq j_2 \leq d_2 - 1$, $P_u(t) \equiv t^{j_1}P_v(t) \pmod{P_1(t)}$, $P_u(t) \equiv t^{j_2}P_v(t) \pmod{P_2(t)}$, and $j_1 \equiv j_2 \pmod{\gcd(d_1, d_2)}$, where d_1 is the order of $P_u(t)$ and $P_v(t)$ with respect to $P_1(t)$ and d_2 is the order of $P_u(t)$ and $P_v(t)$ with respect to $P_2(t)$.

Proof: This is merely a special case of Theorem I-3-3, since the set of of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n.

Theorem II-1-11: Let P(t) be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that P(t) is of period $n = \text{lcm}(n_1, n_2)$. Let D be the set of all spot-errors of diameter $\leq$ a and let each non-zero polynomial $P_u(t)$ in D* be of order n_1 with respect to $P_1(t)$. Finally, suppose that for each distinct non-zero normal form v in D there exists a distinct polynomial $R_{2v}(t)$ which is the representative polynomial of $P_v(t)$ with respect to $P_2(t)$ and is of order n_2 with respect to $P_2(t)$. Then D is of order n and the code C generated by P(t) corrects D.

Proof: This is merely a special case of Theorem I-3-4, since the set of all spot-errors of diameter $\leq a$ is a set of cyclic errors of length n.

Corollary: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that $P(t)$ is of period $n = \text{lcm}(n_1, n_2)$. Let D be the set of all spot-errors of diameter $\leq a$ and suppose that $P_1(t)$ is irreducible and divides no non-zero polynomial $P_u(t)$ in D*. Finally, suppose that for each distinct non-zero normal form v in D there exists a distinct polynomial $R_{2v}(t)$ which is the representative polynomial of $P_v(t)$ with respect to $P_2(t)$ and is of order n_2 with respect to $P_2(t)$. Then D is of order n and the code C generated by $P(t)$ corrects D.

Proof: This is merely a special case of the Corollary of Theorem I-3-4, since the set of all spot-errors of diameter $\leq a$ is a set of cyclic errors of length n.

Theorem II-1-12: Let $P(t)$ be any polynomial of the form $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ and $P_2(t)$ are relatively prime and are of period n_1 and n_2 respectively, so that $P(t)$ is of period $n = \text{lcm}(n_1, n_2)$. Let R_2 be a raster of length n_2, let each error of diameter $\leq a$ on R_2 be of order n_2, and suppose that the code C_2 with generating polynomial $P_2(t)$ is a spot-correcting code of diameter a on R_2. Let R be a raster of length n and suppose that the set S of normal forms of the errors of diameter $\leq a$ on R is in one-to-one correspondence with some subset S_2 of normal forms of the errors of diameter $\leq a$ on R_2 as follows: to each non-zero normal polynomial $P_{u_2}(t)$ of diameter $\leq a$ we first add some multiple $Q_2(t)P_2(t)$ of $P_2(t)$ of degree $\leq n - 1$, then form the representative polynomial $P_v(t)$ of $P_u(t) = P_{u_2}(t) + Q_2(t)P_2(t)$ with respect to $t^n + 1$, so that v is the normal form of u. Finally, suppose that each $P_v(t)$ so formed is of order n_1 with respect to $P_1(t)$. Then the code generated by $P(t)$ is a spot-

correcting code of diameter a on R.

Proof: This follows directly from Theorem I-3-5, since the set of all spot-errors of diameter $\leq$ a is a set of cyclic errors of length n, and C_2 corrects any subset of the set of errors D_2 if it corrects D_2.

Section II-2. Correction of Cyclic Spot-Errors on Rasters Determined by Matrix Pairs.

Up to this point, we have not supposed that a raster has any properties other than those given in Definition II-1-1; that is, it must be a set of points in a space E on which are defined an enumeration and a particular kind of distance function. We shall now consider the problem of actually constructing point-sets which satisfy Definition II-1-1 and to which the theorems of Section II-1 can be applied.

The process by which a raster of length n is constructed is briefly as follows: we first define an infinite set of points in N-dimensional Euclidean space by means of a configuration matrix L. We then define a partition of this point set into n equivalence classes by means of an enumeration vector δ_1, and an alignment matrix M, and finally define a distance function $m(r_i, r_j)$ between any two equivalence classes r_i and r_j of points in our original point set. The set $R = (r_0, r_1, \ldots, r_{n-1})$ of equivalence classes $r_0, r_1, \ldots, r_{n-1}$ is then a set of n "points" which constitute a raster according to Definition II-1-1.

The reason for adopting this rather complicated procedure is that we wish to consider cyclic codes defined on finite rasters in N-dimensional Euclidean space. By grouping the points of an infinite point set in N-dimensional Euclidean space into n equivalence classes, we obtain a finite raster on which the distance function behaves cyclicly, as desired, and yet can be defined in

Euclidean terms.

Definition II-2-1: Let E be an N-dimensional Euclidean space in which is defined a Cartesian coordinate system. A non-singular N x N matrix L is the configuration matrix of an infinite set of points F iff the coordinates of each point ξ in F are given by $\xi = L\alpha$, where α is a vector of integers.

We shall henceforth consider only points sets F which may be obtained by a linear transformation of the integer points of a Cartesian coordinate system. The configuration matrix specifies this linear transformation and thus determine whether F is square, rectangular, hexagonal, etc. Although a given configuration matrix defines a unique configuration, the following theorem shows that the matrix defining a given configuration is not unique.

Theorem II-2-1: Let E be an N-dimensional Euclidean space in which is defined a Cartesian coordinate system and an infinite point set F with configuration matrix L. Then L' is also a configuration matrix of F iff there exists a matrix T of integers with determinant 1 such that L' = LT.

Proof: Suppose that L is the configuration matrix of F, that $\xi = L\alpha$ is any point of F, and that T is a matrix of integers with determinant 1. Then there exists a β such that $\xi = LT\,\beta = L'\beta$; since, therefore, $\beta = T^{-1}\alpha$, where α is a vector of integers, we know that β must also be a vector of integers. Therefore, L' = LT is also a configuration matrix of F.

On the other hand, suppose that L and L' are both configuration matrices of F and that ξ is a point of F. Then there exist vectors of integers α and β such that $L\alpha = L'\beta$, which implies that $\alpha = L^{-1}L'\beta$; which implies finally that $L^{-1}L' = T$, and therefore that L' = LT, where T must be a vector of integers with determinant 1, since α and β are vectors of integers.

Definition II-2-2: Let M be an N x N matrix of integers M of the form

$$\begin{bmatrix} n & k_{12} & k_{13} & \cdots & k_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

, where $0 \leq k_{ij} \leq n - 1$, and let δ_1 be the N-vector $\begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$. Then M is said to be an alignment matrix of length n with enumeration vector δ_1.

Theorem II-2-2: Let E be an N-dimensional Euclidean space in which is defined a Cartesian coordinate system and an infinite set of points F with configuration matrix L, let δ_1 be the column N-vector $\begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$, and let M be an alignment matrix of length n. Then every point in F can be represented uniquely in the form $\xi = iL\,\delta_1 + LM\beta$, where $0 \leq i \leq n - 1$ and β is a vector of integers.

Proof: Let ξ be any point in F. Then there exists a vector of integers α such that $\xi = L\alpha$; and since L is non-singular, $\xi = iL\,\delta_1 + LM\beta$ iff $\alpha = i\,\delta_1 + M\beta$. Recalling the definitions of M and δ_1, we see that this implies that

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \alpha_N \end{bmatrix} = i \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \begin{bmatrix} n & k_{12} & k_{13} & \cdots & k_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \beta_N \end{bmatrix}$$

Rewriting this matrix equation as a set of N simultaneous linear equations, we have:

$$\alpha_1 = i + n\,\beta_1 + \sum_{j=2}^{N} k_{1j}\,\beta_j$$
$$\alpha_2 = \beta_2$$
$$\vdots$$
$$\alpha_N = \beta_N$$

Since $\sum_{j=2}^{N} k_{1j}\,\beta_j$ is a constant, there exist unique integers i and β_1 such that $0 \leq i \leq n-1$ and $\beta_1 = \dfrac{\alpha_1 - i - \sum_{j=2}^{N} k_{1j}\,\beta_j}{n}$. Letting $\beta_j = \alpha_j$ for $j = 2, \ldots, N$, we have

$$\xi = L\alpha = iL \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + LM \begin{bmatrix} \alpha_1 - i - \sum_{j=2}^{N} k_{1j}\,\alpha_j \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_N \end{bmatrix} = iL\,\delta_1 + LM\beta,$$

where $0 \leq i \leq n-1$ and β is a vector of integers.

Definition II-2-3: Let β and γ be N-vectors and let M be an alignment matrix of length n. We say that $\beta \equiv \gamma \pmod{M}$ iff there exists a vector of integers α such that $\beta - \gamma = M\alpha$.

Theorem II-2-3: Let E be an N-dimensional Euclidean space in which is defined a Cartesian coordinate system and an infinite point set F with configuration matrix L, and let M be an alignment matrix of length n with enumeration vector δ_1. Then M partitions the set F into n equivalence classes under the equivalence relation $\xi \equiv \zeta \pmod{LM}$.

Proof: By Theorem II-2-2, we know that there exist an integer i and a vector of integers α such that $\xi = iL\,\delta_1 + LM\,\alpha$, where $0 \leq i \leq n-1$, and we know that there exist an integer j and a vector of integers β

such that $\zeta = jL\ \delta_1 + LM\ \beta$, where $0 \leq j \leq n - 1$. Since $\xi \equiv \zeta \pmod{LM}$ iff there exists a vector of integers γ such that $\xi - \zeta = LM\ \gamma$, we see that $\xi - \zeta = (i - j)L\ \delta_1 + LM(\alpha - \beta) = LM\ \gamma$ implies that $i \equiv j \pmod{n}$. Since i must be an integer such that $0 \leq i \leq n - 1$, we see that M partitions the set F into n equivalence classes under the equivalence relation $\xi \equiv \zeta \pmod{M}$.

Corollary: If M is an N x N alignment matrix of length n and δ_1 is the column vector $\begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$, then $i\ \delta_1 \equiv 0 \pmod{M}$ iff $i \equiv 0 \pmod{n}$.

Proof: It is clear from Theorem II-2-2 that $i\ \delta_1$ and 0 are in the same equivalence class iff $i \equiv 0 \pmod{n}$.

Definition II-2-4: The equivalence class of points ξ in F such that $\xi = iL\ \delta_1 + LM\ \beta$ is called the equivalence class r_i.

The function of the alignment matrix and the enumeration vector is to specify 1) the length n of the raster $R = (r_0, r_1, \ldots, r_{n-1})$ being constructed and 2) the alignment of the equivalence classes of the points of F relative to each other. Specifically, the first column of LM gives the coordinates of the first point belonging to equivalence class r_0 in the $L\ \delta_1$ -- direction from the origin. For $i = 2, \ldots, N$, the ith column of LM gives the coordinates of the first point belonging to equivalence class r_0 in the $L\ \delta_1$-direction from the point $L\ \delta_i$, where δ_i is a column N-vector with i^{th} coordinate equal to 1 and all other coordinates equal to 0.

We now proceed to define a distance function between any two equivalence classes r_i and r_j based on the minimum Euclidean distance between points of

F in the respective equivalence classes.

Definition II-2-5: Let F be an infinite point set which is partitioned by an alignment matrix M with enumeration vector δ_1 into n equivalence classes. The distance $m(r_i, r_j)$ from equivalence class r_i to equivalence class r_j is given by $m(r_i, r_j) = \min_{\mathfrak{z} \varepsilon r_j} d(\mathfrak{x}, \mathfrak{z})$ where $\mathfrak{x}$ is any point in r_i and $d(\mathfrak{x}, \mathfrak{z})$ is the ordinary Euclidean distance from the initial point $\mathfrak{x}$ to the terminal point $\mathfrak{z}$.

The following theorem shows that the value obtained for $m(r_i, r_j)$ does not depend on which point $\mathfrak{x}$ of r_i is chosen as the initial point.

Theorem II-2-4: Suppose that $\mathfrak{x}$ is in r_i, $\mathfrak{z}$ is in r_j, and that $d(\mathfrak{x}, \mathfrak{z}) = m(r_i, r_j)$. Then if $\mathfrak{x}'$ is any other point in r_i, there exists a point $\mathfrak{z}'$ in r_j such that $d(\mathfrak{x}', \mathfrak{z}') = m(r_i, r_j)$.

Proof: Let $\mathfrak{z}'$ be the point $\mathfrak{z}' = \mathfrak{x}' + (\mathfrak{z} - \mathfrak{x})$. Then $d(\mathfrak{x}', \mathfrak{z}') = d(\mathfrak{x}, \mathfrak{z}) = m(r_i, r_j)$ because $\mathfrak{z}' - \mathfrak{x}' = \mathfrak{z} - \mathfrak{x}$. Furthermore, $\mathfrak{z}'$ is in r_j because $\mathfrak{z}' - \mathfrak{z} = \mathfrak{x}' - \mathfrak{x} = LM\beta$, which means that $\mathfrak{z}'$ is in the same equivalence class r_j as $\mathfrak{z}$.

Theorem II-2-5: Let E be an N-dimensional Euclidean space in which is defined a Cartesian coordinate system and an infinite point set F with configuration matrix L, let M be an alignment matrix of length N with enumeration vector δ_1 and let the distance between any two equivalence classes r_i and r_j of F be as given in Definition II-2-5. Then the set R of equivalence classes $r_0, r_1, \ldots, r_{n-1}$ of F is a raster as defined in Definition II-1-1.

Proof: Since R is clearly an ordered n-tuple, it will be necessary to show only that the distance function $m(r_i, r_j)$ satisfies Definition II-1-1.

1) $m(r_i, r_j) = 0$ for each r_i in R since, if $\mathfrak{x}$ is any point in r_i, $m(r_i, r_i) = \min_{\mathfrak{z} \varepsilon r_i} d(\mathfrak{x}, \mathfrak{z}) = d(\mathfrak{x}, \mathfrak{x}) = 0$

2) $m(r_i, r_j) = m(r_j, r_i)$ for all pairs $r_i \neq r_j$, since if ξ is any point in r_i, $m(r_i, r_j) = \min_{\zeta \varepsilon r_j} d(\xi, \zeta) = \min_{\varphi \varepsilon r_i} d(\zeta, \varphi) = m(r_j, r_i)$

3) $m(r_i, r_j) + m(r_j, r_k) \geq m(r_i, r_k)$ for all triples r_i, r_j, r_k since if ξ is any point in r_i, ζ any point in r_j, and φ any point in r_k, we have $d(\xi, \varphi) \leq d(\xi, \zeta) + d(\zeta, \varphi)$

Therefore, $\min_{\varphi \varepsilon r_k} d(\xi, \varphi) \leq \min_{\substack{\zeta \varepsilon r_j \\ \varphi \varepsilon r_k}} \left[(d(\xi, \zeta) + d(\zeta, \varphi))\right]$

This implies that:

$$\min_{\varphi \varepsilon r_k} d(\xi, \varphi) \leq \min_{\zeta \varepsilon r_j} d(\xi, \zeta) + \min_{\varphi \varepsilon r_k} d(\zeta, \varphi)$$

By Theorem II-2-4, we know that it is immaterial which points ξ and ζ are chosen as the initial points of r_i and r_j. Therefore, $m(r_i, r_k) \leq m(r_i, r_j) + m(r_j, r_k)$.

4) $m(r_i, r_j)$ depends only on the value of $i - j \pmod n$, since, if $\xi = iL\,\delta_1 + LM\,\beta$ is any point in r_i, and $\zeta = jL\,\delta_1 + LM\,\gamma$ is a point in r_j such that $d(\xi, \zeta) = m(r_i, r_j)$, then

$\zeta' = \zeta - jL\,\delta_1$ is in r_0, $\xi' = \xi - jL\,\delta_1$ is in $r_{(i-j)(\mathrm{mod}\ n)}$, and $d(\xi', \zeta') = d(\xi, \zeta)$. Therefore, $m(r_i, r_j) = m(r_{(i-j)(\mathrm{mod}\ n)}, r_0)$, which means that $m(r_i, r_j)$ depends only on the value of $(i - j) \pmod n$.

Theorem II-2-5 shows that the set of equivalence classes of F satisfy the definition of a raster as given in Definition II-1-1. Therefore, we may think of each equivalence class of points in F as being itself a "point" in an N-dimensional raster R of length n.

Definition II-2-6: The matrix pair (L,M), where L is a configuration matrix and M is an alignment matrix with enumeration vector δ_1 is said to determine the raster $R = (r_0, r_1, \ldots, r_{n-1})$.

Definition II-2-7: The diameter e of a raster R is the minimum distance between two distinct points of any given equivalence class of points in F, and is given by $e = \min_{\alpha \neq 0} d(0, LM\alpha)$, where α is a vector of integers.

The question naturally arises as to what is the maximum possible value of e for a raster of given length n in a space of dimension N. This is a well-worked out topic in the theory of the geometry of numbers, and Cassels, p. 332, gives limits on e relative to n for each dimension N such that $1 \leq N \leq 10$. For N = 1, 2, 3, and 4, these limits are

N = 1	$e \leq n \det L$
N = 2	$e^2 \leq (2n/\sqrt{3}) \det L$
N = 3	$e^3 \leq \sqrt{2}\ n \det L$
N = 4	$e^4 \leq 2n \det L$

Theorem II-2-6: Let L be an N x N configuration matrix, let M be an N x N alignment matrix of length n, let R be the raster of diameter e determined by the matrix pair (L,M), and let C be the code of length n defined on R and generated by the polynomial $P(t) = t^n + 1$. Then C corrects all spot-errors on R.

Proof: This follows directly from Theorem II-1-7.

Theorem II-2-7: Let L be an N x N configuration matrix, let M_1 and M_2 be N x N alignment matrices of length n_1 and n_2 respectively, so that the pairs (L, M_1) and (L, M_2) determine rasters R_1 and R_2 respectively, and let the set of equivalence classes of R_2 be designated $r_0, r_1, \ldots, r_{n_2 - 1}$. Then $M = M_2 M_1$ is an alignment matrix of length $n = n_2 n_1$ and the pair (L,M) determines a raster $R = (r_0, r_1, \ldots, r_{n-1})$ with the property that if ξ is a point of F in equivalence class r_{i_2} of R_2, then there exists an integer α_1 such that ξ is in equivalence class $r_{i_2 + \alpha_1 n_2}$ of R.

Proof: The following direct calculation shows that if M_1 and M_2 are alignment matrices of length n_1 and n_2 respectively, then $M = M_2M_1$ is an alignment matrix of length $n = n_2n_1$:

$$M_2M_1 = \begin{bmatrix} n_2 & k_{12} & k_{13} & \cdots & k_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} n_1 & h_{12} & h_{13} & \cdots & h_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_2n_1 & h_{12}n_2 + k_{12} & h_{13}n_2 + k_{13} & \cdots & h_{1N}n_2 + k_{1N} \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = M$$

Suppose now that $\mathcal{E} = i_2L\,\delta_1 + LM_2\,\beta_2$ is a point in equivalence class r_{i_2} of R_2. We know by Theorem II-2-2 that there exist a unique integer i, $0 \leq i \leq n - 1$, and a vector of integers β such that $\mathcal{E} = iL\,\delta_1 + LM\beta = iL\,\delta_1 + LM_2M_1\,\beta$. This means that $i_2\,\delta_1 + M_2\,\beta_2 = i\,\delta_1 + M_2M_1\,\beta$; i.e., that $(i - i_2)\,\delta_1 = M_2(\beta_2 - M_1\beta)$. But $(\beta_2 - M_1\beta)$ is a vector of integers α, so our equation becomes

$$\begin{bmatrix} i - i_2 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = \begin{bmatrix} n_2 & k_{12} & k_{13} & \cdots & k_{1N} \\ 0 & 1 & 0 & & 0 \\ 0 & 0 & 1 & & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_N \end{bmatrix}$$

Rewriting this matrix equation as a system of N simultaneous linear equations, we obtain

$$\begin{aligned} i - i_2 &= \alpha_1 n_2 + \sum_{j=2}^{N} k_{1j} \alpha_j \\ \alpha_2 &= 0 \\ \alpha_3 &= 0 \\ &\vdots \\ \alpha_N &= 0 \end{aligned}$$

Therefore, $i - i_2 = \alpha_1 n_2$, which means that ξ is in equivalence class $i_2 + \alpha_1 n_2$ of R.

We now know that if R_2 and R are related as in Theorem II-2-7, then given any error u on R, there exists an error u_2 on R_2 such that r_{i_2} is in u_2 iff there exists an α_1 such that $r_{i_2} + \alpha_1 n_2$ is in u. In other words, if $P_u(t)$ is the image polynomial of u with respect to $t^n + 1$, then $P_{u_2}(t)$ is the image polynomial of u with respect to $t^{n_2} + 1$. If we now make the additional assumptions that the diameter of R_2 is e_2 and that the diameter of u on R is less than $e_2/2$, the following generalizations of Fire's original result can be developed.

Theorem II-2-8: Let L be an N x N configuration matrix, let M_1 and M_2 be alignment matrices of length n_1 and n_2 respectively, so that the pairs (L, M_1), (L, M_2), and $(L, M) = (L, M_2M_1)$ determine rasters R_1, R_2, and R respectively as in Theorem II-2-7, and let R_2 be of diameter e_2. Let u and v be any two spot-errors of diameter $< e_2/2$ on R, let $P_u(t)$ and $P_v(t)$ be the image polynomials of u and v with respect to $t^n + 1$, let $P_{u_2}(t)$ and $P_{v_2}(t)$ be the image polynomials of u and v with respect to $t^{n_2} + 1$, and finally suppose that $P_{u_2}(t) = P_{v_2}(t)$. Then there exists a q such that $P_v(t) = t^{qn_2} P_u(t)$.

UNIVERSITY OF
ILLINOIS LIBRARY

Proof: Because of the way in which R_2 and R were constructed and the assumption that $P_{u_2}(t) = P_{v_2}(t)$, it is possible to write $P_u(t) + P_v(t) = Q(t)(t^{n_2} + 1)$. Let t^i be the term of lowest degree in Q(t). (If there is no such term, then $P_u(t) = P_v(t)$ and the theorem is proved.) Consider all the terms of the form t^i, $t^{i + 2n_2}$, ..., and let $t^{i + qn_2}$ be the first such term which is not in Q(t). Then t^i and $t^{i + qn_2}$ are both in $Q(t)(t^{n_2} + 1)$, which means that they must also both be in the sum $P_u(t) + P_v(t)$. Because of the way in which R_2 and R were constructed, we know that t^i and $t^{i + qn_2}$ must be separated by a distance of at least e_2, which means that they must have originated in different spots. Without loss of generality, we take t^i as a term in $P_u(t)$ and $t^{i + qn_2}$ as a term in $P_v(t)$.

Next consider the two spots $P_v(t)$ and $t^{qn_2}P_u(t)$. They are also congruent mod $t^{n_2} + 1$ and hence their sum is either zero or else contains two terms t^x and t^y separated by a distance of at least e_2, as before. However, by the triangle inequality (Definition II-1-1, Property 3) we see that the sum of the distances from t^x to $t^{i + qn_2}$ and from $t^{i + qn_2}$ to t^y is at least e_2, which violates the assumption that $P_u(t)$ and $P_v(t)$ are of diameter $< e_2/2$. Hence, we must have $P_v(t) = t^{qn_2}P_u(t)$.

Theorem II-2-9: Let L be an N x N configuration matrix, let M_1 and M_2 be alignment matrices of length n_1 and n_2 respectively, so that the pairs (L, M_1), (L, M_2), and $(L, M) = (L, M_2M_1)$ determine rasters R_1, R_2, and R respectively as in Theorem II-2-7, and let R_2 be of diameter e_2. Let S be the set of all normal forms u of diameter $< e_2/2$ on R and let S_2 be the set of all normal forms u_2 of diameter $< e_2/2$ on R_2 such that u_2 is in S_2 iff $P_{u_2}(t)$ is the representative polynomial of u with respect to $t^{n_2} + 1$. Then the set S is in one-to-one correspondence with the set S_2 as follows:

to each non-zero normal form $P_{u_2}(t)$ in S_2, we first add some multiple $Q(t)(t^{n_2} + 1)$ of $(t^{n_2} + 1)$ of degree $\leq n - 1$, then form the representative polynomial $P_w(t)$ of $P_u(t) = P_{u_2}(t) + Q(t)(t^{n_2} + 1)$ with respect to $t^n + 1$, so that w is the normal form of u.

Proof: Let u and v be any two distinct normal forms on R of diameter $< e_2/2$, let $P_{u_2}(t)$ and $P_{v_2}(t)$ be the image polynomials of u and v with respect to $t^{n_2} + 1$, let w_2 and z_2 be the normal forms on R_2 of u_2 and v_2 respectively, and suppose that $P_{w_2}(t) = P_{z_2}(t)$. Then there exists a j_2 such that $P_{u_2}(t) \equiv t^{j_2} P_{v_2}(t) \pmod{t^{n_2} + 1}$, which means, by Theorem II-2-8, that $P_u(t) \equiv t^{j_2 + qn_2} P_v(t) \pmod{t^n + 1}$, which contradicts the hypothesis that u and v are distinct normal forms.

On the other hand, since $P_{u_2}(t) = \text{Res } P_u(t) \pmod{t^{n_2} + 1}$, we see that u_2 is unique for a given u. This, together with the fact that distinct u and v in S give rise to distinct u_2 and v_2 in S_2, means that the sets S and S_2 are in one-to-one correspondence with each other.

Because of the way in which R_2 and R were constructed, the points of R_2 are a subset of the points of R. This together with the cyclic character of the distance function, means that the diameter of each normal form u_2 in S_2 is less than or equal to the diameter of the u in S to which it corresponds.

Finally, because $P_{u_2}(t) = \text{Res } P_u(t) \pmod{t^{n_2} + 1}$, we have $P_u(t) = P_{u_2}(t) + Q(t)(t^{n_2} + 1)$, as required.

Theorem II-2-10: Let L be an N x N configuration matrix, let M_1 and M_2 be N x N alignment matrices of length n_1 and n_2 respectively, so that (L, M_1), (L, M_2), and $(L, M) = (L, M_2M_1)$ determine rasters R_1, R_2 and R respectively as in Theorem II-2-7, and let R_2 be of diameter e_2. If u is any

spot-error on R of diameter $< e_2/2$, then the image polynomial $P_{u_2}(t)$ of u with respect to $t^{n_2} + 1$ is of order n_2 with respect to $t^{n_2} + 1$.

Proof: Suppose that d is the smallest positive integer such that $P_{u_2}(t) \equiv t^d P_{u_2}(t) \pmod{t^{n_2} + 1}$. Then $P_u(t) \equiv t^d P_u(t) \pmod{t^n + 1}$, and, by Theorem II-2-8, $d = qn_2$ for some q. Therefore, $P_{u_2}(t)$ is of order n_2 with respect to $t^{n_2} + 1$, which means that u_2 is of order n_2.

Theorem II-2-11: Let L be an N x N configuration matrix, let M_1 and M_2 be N x N alignment matrices of length n_1 and n_2 respectively, so that (L, M_1), (L, M_2), and $(L, M) = (L, M_2M_1)$ determine rasters R_1, R_2, and R respectively as in Theorem II-2-7 and let R_2 be of diameter e_2. Let $P_1(t)$ be any polynomial which is prime to $t^{n_2} + 1$, is of period n_1^* such that $\text{lcm}(n_1^*, n_2) = n$, and has the property that each spot-error of diameter $\leq a$, where $a < e_2/2$, on R is of order n_1^* with respect to $P_1(t)$. Then $P(t) \equiv (t^{n_2} + 1)P_1(t)$ generates a spot-correcting code C of diameter a on R.

Proof: This follows directly from Theorems II-1-12, II-2-6, II-2-9 and II-2-10.

The preceding theorem is the basis for the Fire Method of constructing error-correcting codes discussed in Sections II-3 and IV-3.

Theorem II-2-12: Let L be an N x N configuration matrix, let M_1 and M_2 be N x N alignment matrices of length n_1 and n_2 respectively, so that (L, M_1), (L, M_2), and $(L, M) = (L, M_2M_1)$ determine rasters R_1, R_2, and R respectively as in Theorem II-2-7, and let R_2 be of diameter e_2. Let $P_2(t)$ be the generating polynomial of a spot-correcting code C_2 of diameter $a < e_2/2$ on R_2 and let $P_1(t)$ be any polynomial which is prime to $P_2(t)$, is of period n_1^* such that $\text{lcm}(n_1^*, n_2) = n$, and has the property that each spot-

error of diameter $\leq$ a on R is of order n_1* with respect to $P_1(t)$. Then $P(t) = P_2(t)P_1(t)$ generates a spot-correcting code C of diameter a on R.

Proof: This follows directly from Theorems II-1-12, II-2-9, and II-2-10, and the fact that $P_2(t)$ divides $t^{n_2} + 1$.

The preceding theorem is the basis for the Combination Method of constructing error-correcting codes discussed in Sections II-3 and IV-4.

Section II-3. Methods for Constructing a Cyclic Spot-Error-Correcting Code on a Given Raster.

Let us now summarize the theory which has been developed and see how to apply it to the problem of actually constructing an error-correcting code. In order to do this, however, we will find the following definitions useful.

Definition II-3-1: Let P(t) be any polynomial which generates a cyclic code of length n. Then P(t) is said to generate a shortened cyclic code of length b for any $b < n$.

The shortened cyclic codes are useful because it may happen that we cannot find a code of exactly the desired length b but we can find a code of length $n > b$. This code is then made to simulate a code of length b by setting digits $x_b, x_{b+1}, \ldots, x_{n-1}$ of code word x equal to zero in the encoder, transmitting only digits $x_0, x_1, \ldots, x_{b-1}$, and re-inserting the missing n-b zeros at the decoder. For a more complete discussion of shortened cyclic codes, see Peterson, p. 158.

Definition II-3-2: Let C be a shortened cyclic code of length b generated by a polynomial P(t) of degree m and let each of the g distinct non-zero normal forms in C be of order $n \geq b$ with respect to P(t). We define the efficiency of C to be $(bg + 1)/2^m$.

It is more reasonable to define the efficiency of a shortened code in

terms of b rather than n, as was done for the full-length codes (see Definition I-4-1), because the number of bits actually transmitted (hence subject to error) is b. We shall ignore the small inaccuracy arising from the fact that some errors which are congruent (mod $t^b + 1$) are not necessarily congruent (mod $t^n + 1$).

Definition II-3-3: Let R be an N-dimensional raster with configuration matrix L and of extent $b_1, b_2, \ldots, b_N$ respectively in each of its N dimensions, and let δ_i, for $i = 2, 3, \ldots, N$, be the column N-vector with i^{th} coordinate equal to 1 and all other coordinates equal to 0. Then R is said to be simple iff each of the points $L\,\delta_i$ of F belongs to equivalence class

$$r_{\prod_{j=1}^{i-1} b_j}.$$

In order to see the significance of this definition, let us examine its meaning in two dimensions. In the two-dimensional case of a raster of width b_1 and height b_2, it means that a simple raster is one in which the point $L\,\delta_2$ belongs to equivalence class r_{b_1}. Thus, as is evident from Figure II-2, each point $i_1 L\,\delta_1 + i_2 L\,\delta_2$ of F is in equivalence class $r_{i_2 b_1 + i_1}$ for each pair (i_1, i_2) such that $0 \leq i_1 \leq b_1 - 1$ and $0 \leq i_2 \leq b_2 - 1$.

From Figure II-2 and the comment following Definition II-2-4, it is evident that the alignment matrix for the simple raster of extent $b_1, b_2, \ldots, b_N$ must be

$$M = \begin{bmatrix} b & b{\cdot}b_1 & b{\cdot}b_1b_2 & \cdots & b{\cdot}\prod_{i=1}^{N-1} b_i \\ 0 & 1 & 0 & \cdots\cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

where $b = \prod_{i=1}^{N} b_i$.

Now that we have shown how to find the alignment matrix M for the raster R on which the code C of diameter a is to be defined, we proceed to the problem of finding a polynomial P(t) which generates C on R. We consider first the Direct Method.

Direct Method: In this method, it is first necessary to find the set of g normal forms of the errors of diameter $\leq a$, either by inspection or by a formal calculation. Next, by the trial and error method described at the end of Section I-2, we search for some polynomial P(t) which is of lowest degree m, is of period $n \geq b$, and generates a spot-correcting code C of diameter a on R. The reason for choosing P(t) to be of lowest degree m is that the efficiency of C is defined to be $(bg + 1)/2^m$, so that the efficiency is maximized by keeping m as small as possible.

Although the Direct Method can be very time-comsuming, it does have the advantage that with sufficient effort we can guarantee that the most efficient possible spot-correcting code of given diameter on a given raster will eventually be found.

Fire Method: This is an extension to N dimensions of the method first used by Fire to find 1-dimensional spot-correcting codes. It is based upon the fact demonstrated in Theorem II-2-11 that one of the factors, $(t^{n_2} + 1)$, in certain code-generating polynomials may be thought of as defining the normal form of the error to be corrected and the other factor, $(P_1(t))$, as defining the location of the error within the code word.

In the Fire Method, as in the Direct Method, it is first necessary to find the set of g normal forms of the errors of diameter $\leq a$ on R, either by inspection or by calculation. Then, using a table such as Table IV-1, we search for the smallest n_2 such that there exists 1) a raster R_2 of length n_2

and diameter $e_2 > 2a$ such that R_2 and R are related as in Theorem II-2-7. Finally, we choose some $P_1(t)$ of lowest degree m_1 such that each non-zero spot-error of diameter $\leq a$ on R is of order n_1 with respect to $P_1(t)$, where $\text{lcm}(n_1, n_2) = n \geq b$. Then we know that $P(t) = (t^{n_2} + 1)P_1(t)$ generates a spot-correcting code of diameter a on R.

The reason for choosing n_2 as small as possible is that going from $P_2(t) = t^{n_2} + 1$ to $P_2(t) = t^{n_2 + 1} + 1$ merely increases the value of n by n_1, while at the same time it increases the degree of P(t) by 1, thus decreasing the overall efficiency of the code.

Combination Method. The Combination Method, based on the results demonstrated in Theorem II-2-12, is the most sophisticated of the three methods, in that it combines the Direct Method with the Fire Method and thus greatly extends the usefulness of code-generating polynomials found by the Direct Method. Actually, the only difference between the Fire Method and the Combination Method is that the polynomial $P_2(t)$ is itself arrived at by the Direct Method instead of being taken of the form $P_2(t) = t^{n_2} + 1$, and is selected from pre-computed tables.

As with the Direct Method and the Fire Method, we first find the set of g normal forms of the errors of diameter $\leq a$ on R, either by inspection or calculation. Then, using a table such as one of Tables IV-2 through IV-6, we search for an n_2 such that there exists 1) a raster R_2 of length n_2 such that R_2 and R are related as in Theorem II-2-7 and 2) a polynomial $P_2(t)$ which generates a spot-correcting code of diameter a on R_2.

No definite rules can be given as to exactly which $P_2(t)$ should be selected from the pre-computed tables to generate a code with given properties except that we want the resulting P(t) to be of as low a degree as possible. However, this is not necessarily achieved by taking $P_2(t)$ to be of as low degree as possible, since there may be a $P_2(t)$ which is of higher degree but

results in a final $P(t) = P_2(t)P_1(t)$ of lower degree. It is therefore possible to recommend only that the search for a $P(t)$ be carried on at least until an efficient $P(t)$ is encountered.

CHAPTER III

ONE-DIMENSIONAL CODES

The first application which we shall make of the theory developed in Chapters I and II is to the one-dimensional or linear case. We shall consider codes obtained by the Fire Method and by a modification of the Combination Method.

Section III-1. The Fire Method.

The first to make systematic use of polynomials of the form $P(t) = (t^{n_2} + 1)P_1(t)$ to generate spot-correcting codes on one-dimensional rasters was Fire. These results are now widely known and are summarized in Petersen, p. 183, but we shall take note only of that portion of them contained in Theorem III-1-1.

Definition III-1-1: A burst-error is any spot-error defined on a one-dimensional raster R.

Each of the theorems in Chapter II relating to spot-errors of diameter $\leq a$ may now be interpreted as applying to burst-errors of diameter $\leq a$ on R. The most important of these theorems is the following:

Theorem III-1-1 (Fire): Let L be the 1 x 1 configuration matrix h and let M be the 1 x 1 alignment matrix n_2. Let R_2 be the one-dimensional raster of length n_2 and diameter $e_2 = hn_2$ determined by the matrix pair (L, M_2) with distance function $m_2(r_{i_2}, r_{j_2}) = h(\mathrm{Res}(j_2 - i_2)(\mathrm{mod}\ n_2))$ and let R be the one-dimensional raster of length $n = n_2 n_1$ determined by the matrix pair $(L, M) = (L, M_2 M_1)$ with distance function $m(r_i, r_j) = h(\mathrm{Res}(j - i)(\mathrm{mod}\ n))$. Let $P_1(t)$ be any polynomial which is prime to $t^{n_2} + 1$, is of period n_1^* such that $l.c.m.(n_1^*, n_2) = n$, and has the property that each burst-error of diameter $\leq a$, where $a < e_2/2$, on R is of order n_1^* with respect to $P_1(t)$. Then $P(t) = (t^{n_2} + 1)P_1(t)$ generates a burst-correcting code C of diameter a on R.

Proof: This follows directly from Theorem II-2-11.

Section III-2. The Modified Combination Method.

We make use of the fact that the set of all polynomials of diameter $\leq hn_2/2$ on a one-dimensional raster R is precisely the set of all polynomials of degree $< n_2/2$ in the following way: instead of using the Direct Method to find polynomials $P_2(t)$ of varying efficiencies which generate spot-correcting codes of given length and diameter and then multiplying by an appropriate $P_1(t)$ as in the regular Combination Method, we use a modification of the Combination Method which finds all polynomials $P(t) = P_2(t)P_1(t)$ which generate maximally efficient codes of given diameter and length. We present first the theoretical basis of the method, then a description of the computational method and finally a listing of numerical results.

It should be noted that these results were arrived at simultaneously and independently by the author and by Elspas and Short. They are given in Foulk and included here because the method used is somewhat different from that used by Elspas and Short and because their results extend only to errors of diameter ≤ 4.

Theorem III-2-1: A cyclic group code of C of length n with generating polynomial P(t) and defined on a one-dimensional raster R is a burst-correcting code of diameter 0 iff P(t) is of period n.

Proof: By Theorem II-1-3, C is a burst-correcting code of diameter 0 iff $t^{i_1} \not\equiv t^{i_2} \pmod{P(t)}$ unless $i_1 \equiv i_2 \pmod{n}$; i.e., iff $t^{i_1} + t^{i_2} \equiv 0 \pmod{P(t)}$ implies that $i_1 - i_2 = kn$ for some k. Therefore, C is a burst-correcting code of diameter 0 iff $t^{i_2}(t^{i_1 - i_2} + 1) \equiv 0 \pmod{P(t)}$ implies $i_1 - i_2 = kn$; i.e. iff P(t) divides no polynomial of the form $t^h(t^j + 1)$, where $1 \leq j \leq n - 1$. We already know that P(t) does not divide t^h for any h and that P(t) must divide $t^n + 1$; therefore, C is a burst-correcting code of diameter 0 iff P(t) is of period n.

Theorem III-2-2: If C is a cyclic group code of length n defined on a one-dimensional raster R and has generating polynomial $P(t) = (t + 1)P_1(t)$, where $P_1(t)$ is of period n and is not divisible by $t + 1$, then C is a burst-correcting code of diameter 1.

Proof: We first let $K_1(t)$ and $K_2(t)$ be any two polynomials of degree ≤ 1 with non-zero constant term, and then show that $t^{i_1}K_1(t) \equiv t^{i_2}K_2(t) \pmod{P(t)}$ implies $K_1(t) = K_2(t)$ and $i_1 \equiv i_2 \pmod{n}$. There are four cases to consider:

Case I: $K_1(t) = 1$, $K_2(t) = 1$. Since P(t) is of period n, we see by Theorem II-2-1 that C is a burst-correcting code of diameter 0; therefore $t^{i_1} \equiv t^{i_2} \pmod{P(t)}$ implies that $i_1 \equiv i_2 \pmod{n}$.

Case II: $K_1(t) = 1$, $K_2(t) = t + 1$. Since P(t) is divisible by $t + 1$, we have $P(t) = (t + 1)P_1(t)$ for some $P_1(t)$. Therefore, $t^{i_1} \equiv t^{i_2}(t + 1) \pmod{P(t)}$ implies $t^{i_1} \equiv t^{i_2}(t + 1) \pmod{t + 1}$, which is clearly impossible since $t^{i_1} \equiv 1 \pmod{t + 1}$ and $t^{i_2}(t + 1) \equiv 0 \pmod{t + 1}$. Therefore, $t^{i_1} \not\equiv t^{i_2}(t + 1) \pmod{P(t)}$.

Case III: $K_1(t) = t + 1$, $K_2(t) = 1$. It follows by symmetry from Case II that $t^{i_1}(t + 1) \not\equiv t^{i_2} \pmod{P(t)}$.

Case IV: $K_1(t) = t + 1$, $K_2(t) = t + 1$. Since $t + 1$ divides P(t), we have $P(t) = (t + 1)P_1(t)$. By Theorem I-3-1, $P_1(t)$ is of period n Therefore $t^{i_1}(t + 1) \equiv t^{i_2}(t + 1) \pmod{P(t)}$ implies $t^{i_1} \equiv t^{i_2} \pmod{P_1(t)}$, which finally implies that $i_1 \equiv i_2 \pmod{n}$.

Taking these four cases together, we see by Theorem II-1-3 that C is a burst-correcting code of diameter 1.

Theorem III-2-3: If there exists an efficient burst-correcting code C of length n, diameter r and with generating polynomial P(t) of degree m, then for C we have

$$\frac{2^{m-r-1} - 1}{} < n \leq \frac{2^{m-r} - 1}{} .$$

Proof: The number of burst errors of diameter $\leq 0, 1, \ldots, r$ are respectively $1 + n, 1 + 2n, \ldots, 1 + n2^r$. Since by Corollary 2 of Theorem I-1-10, $N(C) = 2^{n-m}$, we know by Theorems I-1-1 and I-4-2 that

$$2^{m-1} < 1 + n2^r \leq 2^m$$

$$2^{m-1} - 1 < n2^r \leq 2^m - 1$$

$$2^{m-r-1} - 1/2^r < n \leq 2^{m-r} - 1/2^r$$

Recalling, however, that n must be an integer, we obtain finally

$$2^{m-r-1} - 1 < n \leq 2^{m-r} - 1.$$

Theorem III-2-4: A burst-correcting code of length n, diameter r, and with associated number m is maximally efficient iff $n = 2^{m-r} - 1$.

Proof: By Theorem I-4-3, C is maximally efficient iff $n2^r + 1 > 2^m - 2^r$; i.e., $n > 2^{m-r} - 1 - 1/2^r$. Recalling again that n must be an integer $\leq 2^{m-r} - 1$, we obtain finally $n = 2^{m-r} - 1$.

Theorem III-2-5: Let C be a burst-correcting code of diameter r and with associated number m, and let D be the set of all burst errors of diameter $\leq r$. Then the pair (C,D) is close-packed iff C is a maximally efficient code of diameter 0.

Proof: By Theorem I-1-7, (C,D) is close-packed iff $N(D) = 1 + n2^r = 2^m$; i.e., iff $n = \frac{2^m - 1}{2^r}$. Since n must be an integer, this is possible iff $r = 0$. Therefore, (C,D) is close-packed iff $n = 2^m - 1$; i.e., iff C is maximally efficient of diameter 0.

Theorem III-2-6: (Similar to a theorem of N. M. Abramson). If C is a maximally efficient binary burst-correcting code of diameter r, with generating polynomial P(t) of degree m and if $0 \leq r \leq \frac{m}{3}$, then P(t) is of the form $P_1(t)P_2(t)$, where $P_1(t)$ is primitive of degree $m - r$ and $P_2(t)$ is some polynomial of degree r which belongs to a divisor of $2^{m-r} - 1$.

Proof: Since C is maximally efficient, we know that C is of length $n = 2^{m-r} - 1$, and it is therefore true that P(t) belongs to $2^{m-r} - 1$.

Let Q(t) be any factor of P(t). Assume Q(t) is of degree h. We begin by proving that either $h \leq r$ or $h \geq m - r$. In a binary burst-correcting code of diameter r the number of distinct polynomials of the form Res $t^i K(t)$ (mod P(t)), where K(t) is of degree $\leq r$ with non-zero constant term is $n2^r$ since each choice of $i = 0, 1, \ldots, n - 1$ and each choice of K(t) of degree $\leq r$ with non-zero constant term must yield a distinct residue. Hence for a maximally efficient binary code we obtain just $2^m - (2^r(2^{m-r} - 1)) = 2^r$ polynomials of degree $< m$ which are not of the form Res $t^i K(t)$ (mod P(t)). However, if $h > r$, then any polynomial of the form R(t)Q(t) is not of the form Res $t^i K(t)$. For if we assume $R(t)Q(t) \equiv t^i K(t) \pmod{P(t)}$, then $t^{n-i} R(t)Q(t) \equiv K(t) \pmod{P(t)}$, and since P(t) is divisible by Q(t), this would imply that K(t) must also be divisible by Q(t). Since we have just 2^{m-h} choices for R(t), it follows that $2^{m-h} \leq 2^r$ and hence $r \geq m - h$.

We next prove that P(t) has at least one irreducible factor $P_1(t)$ of degree $\geq m - r$. Assume first that $r < \frac{m}{3}$, and that no irreducible factors of P(t) of degree $\geq m - r$ exist. Hence all irreducible factors of P(t) are of degree $\leq r$. Let $P(t) = Q_1(t)Q_2(t)$ be any decomposition of P(t) into two factors. By the previous result, one of these factors, say $Q_1(t)$, is of degree $\leq r$. Since $Q_2(t)$ is of degree $\geq m - r$, it must be composite. Hence, we may pick any factor $P_i(t)$ of $Q_2(t)$ of degree $\leq r$ and write $Q_1^*(t) = P_i(t)Q_1(t)$, and $Q_2^*(t) = Q_2(t)/P_i(t)$. Thus $P(t) = Q_1^*(t)Q_2^*(t)$ and $Q_1^*(t)$ is of degree $\leq 2r <$

$m - r$ and hence Q_1^* is of degree $\leq r$. However, this process may be repeated indefinitely, each time increasing the degree of $Q_1(t)$ until its degree exceeds r, thus giving a contradiction.

In the case $r = \frac{m}{3}$, then by the above argument P(t) must consist solely of factors of degree r. This means that $P(t) = P_1(t)P_2(t)P_3(t)$, where all three factors are of degree r. This means that all expressions of the form $R(t)P_1(t)P_2(t)$, $R(t)P_1(t)P_3(t)$, and $R(t)P_2(t)P_3(t)$ are <u>not</u> of the form Res $t^i K(t)$ (mod P(t)) but since there are $3 \times 2^m - 2r$ of these polynomials, the code cannot be maximally efficient, for then only 2^r of these polynomials may exist.

We may now write $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ is irreducible of degree $\geq m - r$. We wish to show that $P_1(t)$ is primitive of degree $m - r$. To show this we note that all expressions of the form $t^i P_2(t)$ are incongruent mod P(t) for $i = 0, 1, \ldots, n - 1$. Hence, all t^i are incongruent mod $P_1(t)$ and $P_1(t)$ therefore belongs to n. Thus, if h is the degree of $P_1(t)$ we see that $n = 2^{m-r} - 1$ divides $2^h - 1$ so that $m - r$ divides h. Hence, since $m - r \leq h \leq m$, we have $h = m - r$ and $P_1(t)$ is primitive.

Finally, we note that since $t^n \equiv 1$ (mod P(t)) we must have $t^n \equiv 1$ (mod $P_2(t)$) so $P_2(t)$ must belong to a divisor of $2^{m-r} - 1$.

<u>Theorem III-2-7</u>: If C is a binary cyclic group code with generating polynomial P(t) of degree $m \geq 2$, then C is a maximally efficient burst-correcting code of diameter 0 iff P(t) is primitive.

Proof: By Theorem III-2-1, we know that C is burst-correcting of diameter 0 iff the period of P(t) is the same as the length n of C. Since C is maximally efficient iff $n = 2^m - 1$ and P(t) is primitive iff it belongs to $2^m - 1$, it follows that C is maximally efficient iff P(t) is primitive.

Theorem III-2-8: If C is a binary cyclic group code with generating polynomial P(t) of degree $m \geq 3$, then C is a maximally efficient burst-correcting code of diameter 1 iff P(t) is of the form $P(t) = (t + 1)P_1(t)$, where $P_1(t)$ is primitive of degree $m - 1$.

Proof: First, suppose that C is a maximally efficient burst-correcting code of diameter 1. Then, by Theorem III-2-6, we know that P(t) is of the form $P_2(t)P_1(t)$, where $P_1(t)$ is primitive of degree $m - 1$ and $P_2(t)$ is a polynomial of degree 1 which belongs to a divisor of $2^{m-1} - 1$. Since the only polynomial of degree 1 which belongs to a divisor of $2^{m-1} - 1$ is $t + 1$, we see that $P(t) = (t + 1)P_1(t)$.

Conversely, suppose that P(t) is of the form $P(t) = (t + 1)P_1(t)$, where $P_1(t)$ is primitive of degree $m - 1$. By Theorem III-2-2, C is a burst-correcting code of diameter 1 and length $2^{m-1} - 1$; therefore, C is a maximally efficient burst-correcting code of diameter 1.

Theorem III-2-9: Let C be a maximally efficient binary burst-correcting code of length $n = 2^{m-r} - 1 = 2^s - 1$ and diameter r with generating polynomial P(t) of degree m, where $0 \leq r \leq \frac{m}{3}$, so that $P(t) = P_2(t)P_1(t)$, where $P_1(t)$ is primitive of degree s and $P_2(t)$ is of degree r and belongs to divisor of $2^s - 1$. Also, let $P_2(t) = P_{21}(t)P_{22}(t)$, where $P_{21}(t)$ is of degree r_1, $P_{22}(t)$ is of degree r_2, and $P_{21}(t)$ and $P_{22}(t)$ are relatively prime. Then $Q_1(t) = P_{21}(t)P_1(t)$ is the generating polynomial of degree $m_1 = r_1 + s$ of a maximally efficient binary burst-correcting code C_1 of length $2^s - 1$ and diameter r_1 and $Q_2(t) = P_{22}(t)P_1(t)$ is the generating polynomial of degree $m_2 = r_2 + s$ of a maximally efficient binary burst-correcting code C_2 of length $2^s - 1$ and diameter r_2.

Proof: Since C is a maximally efficient burst-correcting code of length $2^s - 1$ and diameter r, we know by Theorem II-1-3 that $t^{i_1}K_1(t) \equiv$

$t^{i_2}K_2(t) \pmod{P(t)}$ implies that $K_1(t) = K_2(t)$ and $i_1 \equiv i_2 \pmod{2^s - 1}$ if $K_1(t)$ and $K_2(t)$ are any polynomials of degree $\leq r$ with non-zero constant term. Suppose now that $K_{11}(t)$ and $K_{21}(t)$ are any polynomials of degree $\leq r_1$ with non-zero constant term and that $t^{i_{11}}K_{11}(t) \equiv t^{i_{21}}K_{21}(t) \pmod{Q_1(t)}$. Then $t^{i_{11}}K_{11}P_{22}(t) \equiv t^{i_{21}}K_{21}(t)P_{22}(t) \pmod{P(t)}$, which implies that $K_{11}(t)P_{22}(t) = K_{21}(t)P_{22}(t)$ and also that $i_{11} \equiv i_{21} \pmod{2^s - 1}$, which implies finally that $K_{11}(t) = K_{21}(t)$. Therefore, $Q_1(t)$ is the generating polynomial of degree $m_1 = r_1 + s$ of a maximally efficient binary burst-correcting code C_1 of length $2^s - 1$ and diameter r_1. Similarly, $Q_2(t)$ is the generating polynomial of degree $m_2 = r_2 + s$ of a maximally efficient binary burst-correcting code C_2 of length $2^s - 1$ and diameter r_2.

Theorem III-2-9 can save us a considerable amount of labor when we are searching systematically for polynomials which generate maximally efficient burst-correcting codes if our search is organized so that we test $Q_1(t)$ and $Q_2(t)$ before testing $P(t)$. If we find that either $Q_1(t)$ fails to generate a maximally efficient burst-correcting code of diameter r_1 or $Q_2(t)$ fails to generate a maximally efficient burst-correcting code of diameter r_2, then we do not need to test $P(t)$ since we know that it cannot generate a maximally efficient burst-correcting code of diameter r.

Section III-3. The Computer Search.

The Illiac high-speed computer has been used to search for polynomials $P_1(t)$ and $P_2(t)$ whose products $P(t) = P_2(t)P_1(t)$ generate maximally efficient burst-correcting codes of various lengths and diameters. The procedure actually used is one suggested in a private communication from C. W. Gear of the University of Illinois Digital Computer Laboratory. This procedure is as follows:

For each polynomial $K(t)$ of degree $\leq r$ with non-zero constant term, we define $A_1(K(t))$ to be the smallest positive integer such that $t^{A_1(K(t))} \equiv K(t) (\mathrm{mod}\ P_1(t))$. Since $P_1(t)$ is primitive, such an integer always exists. If $P_2(t)$ is also primitive, then we similarly define $A_2(K(t))$ to be the smallest positive integer such that $t^{A_2(K(t))} \equiv K(t) (\mathrm{mod}\ P_2(t))$. If $P_2(t)$ is not primitive, however, this definition will not serve since there may not be any h such that $t^h \equiv K(t) (\mathrm{mod}\ P_2(t))$. In such a case, we order the polynomials $K_0(t)$, ..., $K_{2^r - 1}(t)$ of degree $\leq r$ with non-zero constant term in any way such that $K_0(t) = 1$ is first. Then we define recursively:

(1) $A_2(1) = 0$

For each $K_i(t)$, $i = 1, \ldots, 2^r - 1$, $A_2(K_i(t))$ is defined as follows:

(2) If there exists a $K_j(t)$ with $j < i$ such that $A_2(K_j(t))$ has already been defined to be zero, and there exists an integer h such that $t^h K_j(t) \equiv K_i(t) (\mathrm{mod}\ P_2(t))$, then $A_2(K_i(t))$ is the smallest positive integer such that $t^{A_2(K_i(t))} K_j(t) \equiv K_i(t) (\mathrm{mod}\ P_2(t))$.

(3) If there exists no such $K_j(t)$ with $j < i$, then define $A_2(K_i(t))$ to be 0.

It is easy to see that if $P_2(t)$ is in fact primitive, this definition reduces to the one given above for primitive $P_2(t)$.

We now have $t^{A_1(K(t)) - A_1(M(t))} M(t) \equiv K(t) (\mathrm{mod}\ P_1(t))$ whenever $K(t)$ and $M(t)$ are any two polynomials of degree $\leq r$ with non-zero constant term, and

we also have $t^{A_2(K(t)) - A_2(M(t))} M(t) \equiv K(t) \pmod{P_2(t)}$ whenever K(t) and M(t) are any two polynomials of degree $\leq$ r with non-zero constant term for which there exists any h such that $t^h M(t) \equiv K(t) \pmod{P_2(t)}$.

The next step in our procedure is to compute $A_2(K(t)) - A_2(M(t))$ for each distinct pair of polynomials K(t) and M(t) such that there is an integer h having the property that $t^h K(t) \equiv M(t) \pmod{P_2(t)}$. This means that $t^{A_2(K(t))} M(t) \equiv t^{A_2(M(t))} K(t) \pmod{P_2(t)}$ because of the manner in which the function A_2 was formed. We next compute $A_1(K(t)) - A_1(M(t))$ for each such pair of polynomials and then determine whether or not $A_1(K(t)) - A_1(M(t)) \equiv A_2(K(t)) - A_2(M(t)) \pmod{d}$, where d is the divisor of $2^{m-r} - 1$ which is the period of $P_2(t)$. The above congruence fails to hold for every such pair K(t) and M(t) iff P(t) generates a code, which result may be derived directly from Theorem II-1-3.

This procedure was carried out on Illiac for all combinations of polynomials $P_2(t)$ and $P_1(t)$ such that $P_1(t)$ is primitive of degree $s = m - r \leq 13$, the period of $P_2(t)$ is a divisor of $2^{m-r} - 1$ and is of degree $\leq s$, and such that it has not previously been found that the product of some factor of $P_2(t)$ with $P_1(t)$ fails to give a code-generating polynomial. By Theorem III-2-6, we know that we have now found <u>all</u> maximally efficient burst-correcting codes such that $s = m - r \leq 13$ and $r \leq \frac{m}{3}$. If Illiac found that $P(t) = P_2(t)P_1)t)$ was a code-generating polynomial, it indicated this by printing out an "OK". If, however, P(t) failed to generate a code, then Illiac indicated that fact by printing out the causes of the failure. That is, for each combination of K(t) and M(t) such that $t^{A_1(K(t)) - A_1(M(t))} M(t) \equiv K(t) \pmod{P(t)}$ with $M(t) \neq K(t)$, Illiac printed out M(t), K(t), $A_1(K(t)) - A_1(M(t))$, and, if this was the first occurrence of M(t) for this P(t), $A_1(M(t))$ was also printed out. The complete results are too bulky to be included in this paper, but they are summarized in Table III-1, where each polynomial $P(t) = P_2(t)P_1(t)$ which

generates a code is designated by an "0" in the corresponding square and each P(t) which fails to generate a code is designated by an "x" in the corresponding square.

Actually, corresponding to each primitive polynomial $P_1(t)$ included in the table, there is another primitive polynomial $P_1^*(t)$ which may be obtained from $P_1(t)$ by reversing the order of its coefficients; i.e., $P_1^*(t) = t^s P_1(t^{-1})$. However, it is easy to see that each of the polynomials $P_2(t)P_1^*(t)$ would have been found to generate a code iff $P_2^*(t)P_1(t)$ was found to generate a code; therefore, it is not necessary to consider the $P_1^*(t)$'s separately as long as both $P_2(t)$ and $P_2^*(t)$ are included among our $P_2(t)$'s. Note that while we may have $P_2^*(t) = P_2(t)$; due to the fact that $P_1(t)$ is primitive, we never have $P_1(t) = P_1^*(t)$.

Figure III-1 shows what the output from Illiac looks like. In this output, binary information is represented sexadecimally according to the following code:

Binary	Sexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	+
1011	-
1100	N

Binary	Sexadecimal
1101	J
1110	F
1111	L

```
Line 1.     P2 = 90000
     2.     000003
     3.     01 0_ 0J
     4.     000003
     5.     03 0L 05
     6.     000001
     7.     07
     8.     N = 0000000006   PSI N = 0000000006
     9.     P1 = 86000  P2P1 = 96N0000000
    10.     0 - 0J 000013  000035

    11.     P1 = NF000   P2P1 = J7N00  00000
    12.     01  0- 000019 000000

    13.     P1 = J+000    P2P1  N140000000
    14.     03    0L    000016    000008
    15.     03    05    000008
```

Figure III-1

Therefore, Line 1 tells us that $P_2(t)$ is 1001 0000 0000 0000 0000, which is the polynomial $1 \cdot t^3 + 0 \cdot t^2 + 0 \cdot t + 1 \cdot 1$ or $t^3 + 1$. Lines 2 and 3 tell us that the set of polynomials arising from $K_0(t) = 1$ has three members, and specifically that $t \equiv t^3 + t + 1 \pmod{t^3 + 1}$ and $t^2 \equiv t^3 + t^2 + 1 \pmod{t^3 + 1}$. Lines 4 and 5 tell us that there are also three polynomials in the set arising from $K_1(t) = t + 1$ and that $t(t + 1) \equiv t^3 + t^2 + t + 1 \pmod{t^3 + 1}$ and that $t^2(t + 1) \equiv t^2 + 1 \pmod{t^3 + 1}$. Lines 6 and 7 tell us that there is exactly one polynomial in the set arising from $K_2(t) = t^2 + t + 1$. Line 8 tells us that $N(= m - r) = 6$ is the degree of the $P_1(t)$'s which we shall consider (disregard PSI N). Line 9 tells us that $P_1(t)$ is 1000 0110 0000 0000 0000, which is the polynomial $t^6 + t + 1$, and that $P(t) = P_2(t)P_1(t)$ is 1001 0110 110-------0 which is the polynomial $t^9 + t^6 + t^4 + t^3 + t + 1$. Line 10 tells us that P(t) fails to generate a code because

$t^{13}(t^3 + t + 1) \equiv t^3 + t^2 + 1 \pmod{P(t)}$, and also tells us incidentally that $A_1(t^3 + t + 1) = 35$. Lines 11 and 12 tell us that $P(t) = (t^3 + 1)(t^6 + t^5 + t^2 + t + 1)$ fails to generate a code because $t^{19} \equiv t^3 + t + 1 \pmod{P(t)}$ and that $A_1(1) = 0$. Lines 13, 14, and 15 tell us $P(t) = (t^3 + 1)(t^6 + t^5 + t^3 + t^2 + 1)$ fails to generate a code because $t^{16}(t + 1) \equiv t^3 + t^2 + t + 1 \pmod{P(t)}$ and $t^8(t + 1) \equiv t^2 + 1 \pmod{P(t)}$ and that $A_1(t + 1) = 8$. As there are no other primitive polynomials of degree 6 except a $P_1^*(t)$ corresponding to each $P_1(t)$ already considered, there are no codes generated by $P(t)$'s of the form $(t^3 + 1)P_1(t)$ where $P_1(t)$ is of degree 6.

This procedure not only enables us to test a given pair of polynomials $P_2(t)$ and $P_1(t)$ to determine whether or not their product $P(t)$ generates a code but also enables us to estimate the probability that a given $P(t)$ will actually generate a code, assuming that all residues 0, 1, ..., d-1 are equally probable for any given $A_1(K(t))$. While there is no known theoretical basis for assuming that all such residues are equally probable, our results are not inconsistent with such an assumption. To estimate the probability that $P(t)$ will generate a code, we first note that due to the existence of certain relations among the polynomials of degree $\leq$ r with non-zero constant term, there are not actually 2^r independent cases to consider. For example, when testing for codes of level 3 generated by $P(t) = (t^2 + t + 1)P_1(t)$, we first note that in this case d = 3 and that in accordance with the definitions given above, $A_2(1) = 0$, $A_2(t + 1) = 2$, $A_2(t^2 + 1) = 1$, and $A_2(t^2 + t + 1) = 0$. However, if $K(t)$ is any polynomial of degree $\leq$ 2 with non-zero constant term other than $t^2 + t + 1$, then there is no h such that $t^h(t^2 + t + 1) \equiv K(t) \pmod{t^2 + t + 1}$. In other words, since $t^2 + t + 1$ is itself a $K_j(t)$ and is in fact the only member of the set of polynomials arising from $K_j(t) = t^2 + t + 1$, the value of the quantity

$A_1(K(t)) - A_1(t^2 + t + 1) \pmod 3$ for any $K(t) \neq t^2 + t + 1$ is immaterial to the existence of a code. Furthermore, since $A_1(1)$ is always equal to zero and $A_1(t^2 + 1) = 2A_1(t + 1)$ since $t^2 + 1 = (t + 1)^2$ in GF(2), we see that the existence or non-existence of a code may actually be inferred from the value of $A_1(t + 1)$ alone or from the value of $A_1(t^2 + 1)$ alone. Specifically, a code exists iff either $A_1(t + 1) \equiv 0 \pmod 3$ or $A_1(t + 1) \equiv 1 \pmod 3$, whereas a code fails to exist iff $A_1(t + 1) \equiv 2 \pmod 3$. Or, if we prefer to look at $A_1(t^2 + 1)$, we see that a code exists iff either $A_1(t^2 + 1) \equiv 0 \pmod 3$ or $A_1(t^2 + 1) \equiv 2 \pmod 3$, whereas a code fails to exist iff $A_1(t^2 + 1) \equiv 1 \pmod 3$. From this discussion, we may conclude that the probability that $P(t) = (t^2 + t + 1)P_1(t)$ will generate a code is 2/3, which is not inconsistent with the fact that out of 228 $P_1(t)$'s, 150 gave code-generating $P(t)$'s.

As another example, we may estimate the probability that $P(t) = (t^3 + 1)P_1(t) = (t + 1)(t^2 + t + 1)P_1(t)$ will generate a code. Referring to Figure III-1, we see that $P(t)$ fails to generate a code iff one or more of the congruences $A_1(t^3 + t + 1) - A_1(1) \equiv 1 \pmod 3$, $A_1(t^3 + t^2 + 1) - A_1(1) \equiv 2 \pmod 3$, $A_1(t^3 + t^2 + 1) - A_1(t^3 + t + 1) \equiv 1 \pmod 3$, $A_1(t^3 + t^2 + t + 1) - A_1(t + 1) \equiv 1 \pmod 3$, $A_1(t^2 + 1) - A_1(t + 1) \equiv 2 \pmod 3$, or $A_1(t^2 + 1) - A_1(t^3 + t^2 + t + 1) \equiv 1 \pmod 3$ holds. These values may not all be chosen independently, however, since $A_1(t^3 + t^2 + 1) - A_1(t^3 + t + 1) = A_1(t^3 + t^2 + 1) - A_1(1) - (A_1(t^3 + t + 1) - A_1(1))$ and $A_1(t^2 + 1) - A_1(t^3 + t^2 + t + 1) = A_1(t^3 + t^2 + t + 1) - A_1(t + 1) = 2(A_1(t^2 + 1) - A_1(t + 1))$. A straightforward calculation now shows that $P(t)$ generates a code only if: 1) either $A_1(t^3 + t^2 + 1) - A_1(1) \equiv 0 \pmod 3$ and $A_1(t^3 + t + 1) - A_1(1) \equiv 0 \pmod 3$ or $A_1(t^3 + t^2 + 1) - A_1(1) \equiv 1 \pmod 3$ and $A_1(t^3 + t + 1) - A_1(1) \equiv 2 \pmod 3$ and 2) $A_1(t^2 + 1) - A_1(t + 1) \equiv 1 \pmod 3$. Since the first condition can be met in 2 ways out

of 9 possibilities and the second can be met in 2 ways out of 3 possibilities, we may conclude that the probability of obtaining a code-generating polynomial of the form $P(t) = (t^3 + 1)P_1(t)$ is $\frac{2}{9} \times \frac{2}{3} = \frac{4}{27}$, which is not inconsistent with the fact that out of 228 $P_1(t)$'s, 36 gave code-generating polynomials. As a check on our method of estimating the probability of obtaining a code-generating P(t) of this form, we may utilize Theorem III-2-9 and calculate that the probability of obtaining a code-generating P(t) using a $P_1(t)$ which is known to give a code-generating polynomial of the form $(t^2 + t + 1)P_1(t)$ should be $\frac{2}{9}$, which is not inconsistent with the fact that out of 150 such $P_1(t)$'s, 36 gave code-generating P(t)'s.

By a similar calculation, we find that the probability of obtaining a code-generating polynomial of the form $P(t) = (t^3 + t + 1)P_1(t)$ or $P(t) = (t^3 + t^2 + 1)P_1(t)$ should be $\frac{6}{7} \times \frac{2}{7} \times \frac{1}{7} = \frac{12}{343}$, which is not inconsistent with the fact that out of 198 such P(t)'s, 5 were code-generating polynomials.

The probability of finding a code-generating polynomial of the form $P(t) = P_2(t)P_1(t)$ where $P_2(t)$ is of degree 4 or 5 has not been calculated, but it may be expected to be quite low; and, indeed, no such P(t)'s were found for $P_1(t)$ of degree ≤ 13.

It should be emphasized again that these calculations do not give true probabilities, but rather probabilities based on the assumption that the residues of the $A_1(K(t))$'s (mod d) are uniformly distributed.

Section III-4. Summary of One-Dimensional Results.

A computer search has been carried out on Illiac for all combinations of polynomials $P_2(t)$ and $P_1(t)$ such that $P_1(t)$ is primitive of degree $s = m - r \leq 13$, the period of $P_2(t)$ is a divisor of 2^{m-r} and is of degree ≤ 5 and such that it has not previously been found that the product of some factor of $P_2(t)$ with $P_1(t)$ fails to give a code-generating polynomial. Since the cases for $r = 0$ and $r = 1$ are trivial, only the cases where $P_2(t)$ was of degree ≥ 2 were actually run on Illiac. No codes were found except for $r = 2$ and $r = 3$ in the range $s \leq 13$.. Our results are:

1) For $P_2(t) = t^2 + t + 1$, we have

Degree of $P_1(t)$	4	6	8	10	12	Totals
No. of Primitive Polynomials	2	6	16	60	144	228
No. of Codes Found	2	4	10	40	94	150

2) For $P_2(t) = t^3 + 1$, we have

Degree of $P_1(t)$	4	6	8	10	12	Totals
No. of Primitive Polynomials	2	6	16	60	144	228
No. of Codes Found	0	0	0	10	26	36

3) For $P_2(t) = t^3 + t + 1$, we have

Degree of $P_1(t)$	6	9	12	Totals
No. of Primitive Polynomials	6	48	144	198
No. of Codes Found	0	1	4	5

A similar result holds by symmetry for $P_2(t) = t^3 + t^2 + 1$.

By Theorem III-2-6, we know that we have found all maximally efficient binary cyclic burst-correcting codes for which $s = m - r \leq 13$ and $r \leq \frac{m}{3}$.

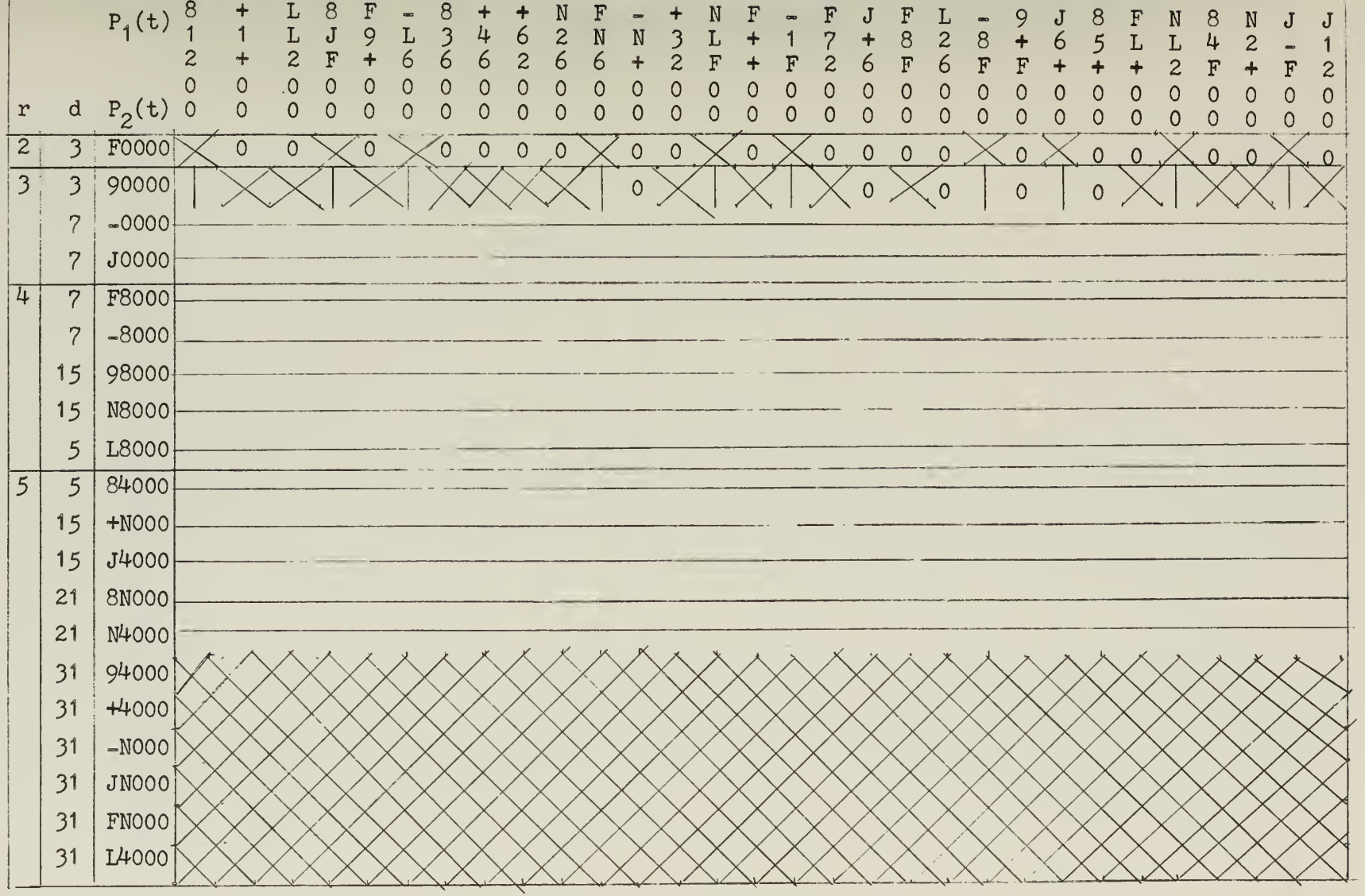
Table III-1 summarizes the results obtained by Illiac in its search for maximally efficient binary cyclic burst-correcting codes of diameter $2 \leq r \leq 5$ and length $\leq 2^{13} - 1$. Since the product $P(t)$ of a pair of polynomials $P_2(t)$ and $P_1(t)$ can be a code-generating polynomial only if the period of $P_2(t)$ divides $2^s - 1$, where s is the degree of $P_1(t)$, a horizontal line has been drawn through the square corresponding to each pair of polynomials such that the period of $P_2(t)$ does not divide $2^s - 1$. Thus, there are no codes of diameter $2 \leq r \leq 5$ and length $2^7 - 1$, $2^{11} - 1$, or $2^{13} - 1$ since each of these numbers is prime to 3, 5, 7, 15, 21, and 31. Also a vertical line has been drawn through the square corresponding to a given $P_2(t)$ and $P_1(t)$ if $P_2(t)$ is composite and it has previously been ascertained that the product of one of the factors of $P_2(t)$ with $P_1(t)$ fails to be a code-generating polynomial. All other squares are marked with an "O" if $P(t)$ is a code-generating polynomial and an "x" if it is not. Each $P_2(t)$ and $P_1(t)$ is expressed in the sexadecimal notation described earlier; thus, $N0000_{(16)} = 1100\ 0000\ 0000\ 0000\ 0000_{(2)}$, which represents the polynomial $t + 1$; similarly F0000 represents $t^2 + t + 1$, -0000 represents $t^3 + t + 1$, 98000 represents $t^4 + t + 1$, etc. Although Theorem III-2-6 applies only when $r \leq \frac{m}{3}$, for the sake of completeness, calculations have been carried out even for those combinations of $P_2(t)$ and $P_1(t)$ for which $r \leq \frac{m}{3}$.

		s	1	2	3	4	5	6	7	8
		$2^s - 1$	1	3	7	15	31	63 = 3·3·7	127	255 = 3·5·17
		$P_1(t)$	N	F	-	9	9 - J	8 N J		8 - L 9 - - + F
			0	0	0	8	4 N N	6 F +		F 4 3 5 2 1 L 1
			0	0	0	0	0 0 0	0 0 0		8 8 8 8 8 8 8 8
			0	0	0	0	0 0 0	0 0 0		0 0 0 0 0 0 0 0
r	d	$P_2(t)$	0	0	0	0	0 0 0	0 0 0		0 0 0 0 0 0 0 0
2	3	F0000				0		0 0		0 0 0 0
3	3	90000								
	7	-0000								
	7	J0000								
4	7	F8000								
	7	-8000								
	15	98000								
	15	N8000								
	5	L8000								
5	5	84000								
	15	+N000								
	15	J4000								
	21	8N000								
	21	N4000								
	31	94000								
	31	+4000								
	31	-N000								
	31	JN000								
	31	FN000								
	31	L4000								

TABLE III-1

r	d	$P_2(t)$	$2^s - 1$ = 9	$P_1(t)$ = 511 = 7·73																								
				8	9	N	N	8	9	J	-	F	L	L	F	J	9	L	J	L	9	L	+	+	L	N	+
				4	6	N	4	-	J	8	6	1	+	8	3	+	-	3	N	2	L	5	5	L	F	5	J
				4	4	4	N	4	N	4	N	4	4	N	N	N	N	4	N	N	4	4	4	4	N	4	N
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	3	F0000																										
3	3	90000																										
	7	-0000																										
	7	J0000																										
4	7	F8000																										
	7	-8000																										
	15	98000																										
	15	N8000																										
	5	L8000																										
5	5	84000																										
	15	+N000																										
	15	J4000																										
	21	8N000																										
	21	N4000																										
	31	94000																										
	31	+4000																										
	31	-N000																										
	31	JN000																										
	31	FN000																										
	31	L4000																										

TABLE III-1 (continued)



r	d	$P_2(t)$	$P_1(t)$	81200	+10F00	LL2F00	8JF900	F9+F00	-L6F00	836F00	+46F00	+62F00	N26F00	FN6F00	-N+F00	+32F00	NLFF00	F++F00	-1F2F00	F72F00	J+6F00	F8FF00	L26F00	-8FF00	9+FF00	J6+F00	85+F00	FL+F00	NL2F00	84FF00	N2+F00	J-F000	J12000
2	3	F0000			0	0		0		0	0	0	0		0	0		0		0	0	0	0		0		0	0		0	0		0
3	3	90000													0						0		0		0		0						
	7	-0000																															
	7	J0000																															
4	7	F8000																															
	7	-8000																															
	15	98000																															
	15	N8000																															
	5	L8000																															
5	5	84000																															
	15	+N000																															
	15	J4000																															
	21	8N000																															
	21	N4000																															
	31	94000																															
	31	+4000																															
	31	-N000																															
	31	JN000																															
	31	FN000																															
	31	L4000																															

TABLE III-1 (continued)

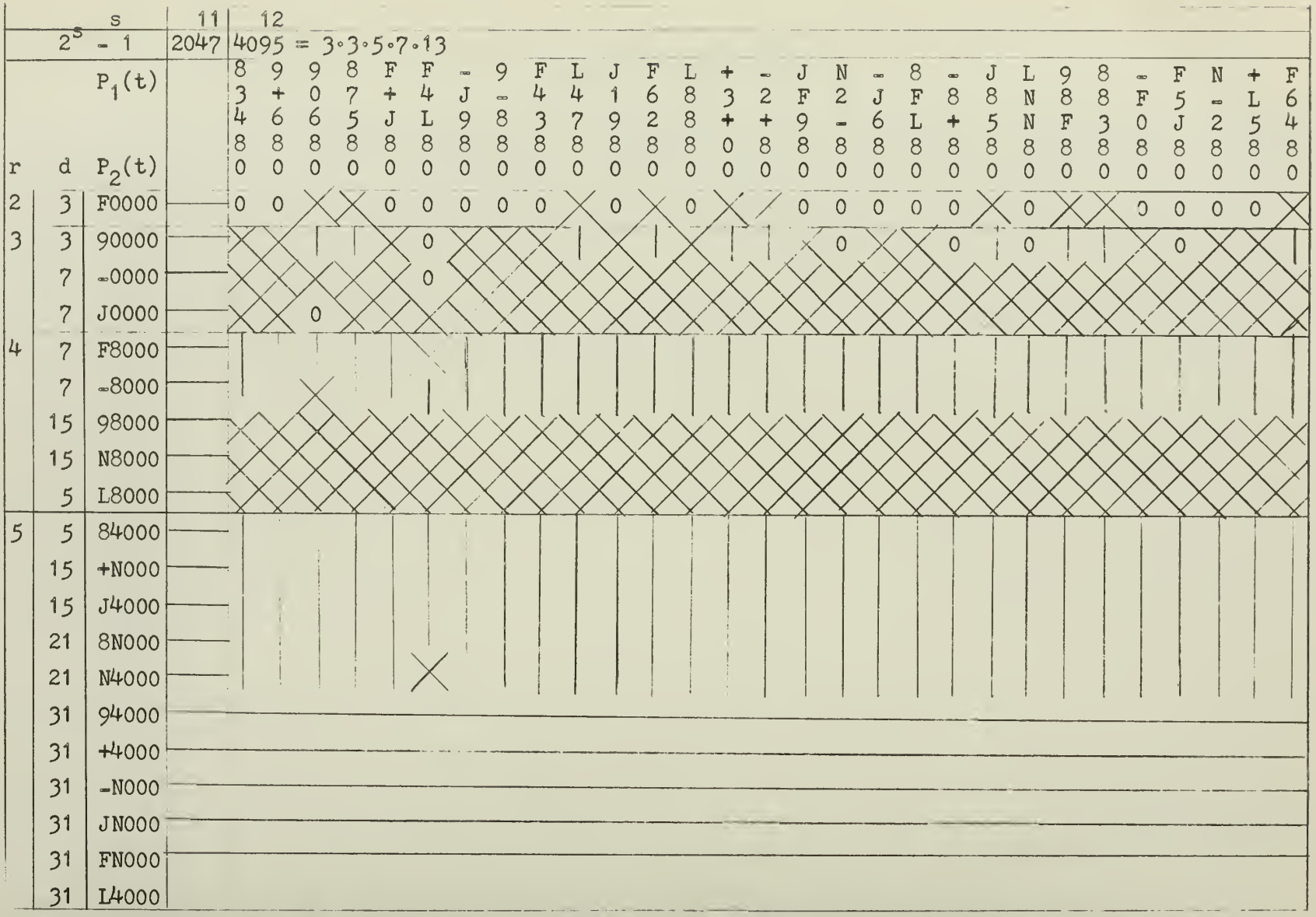


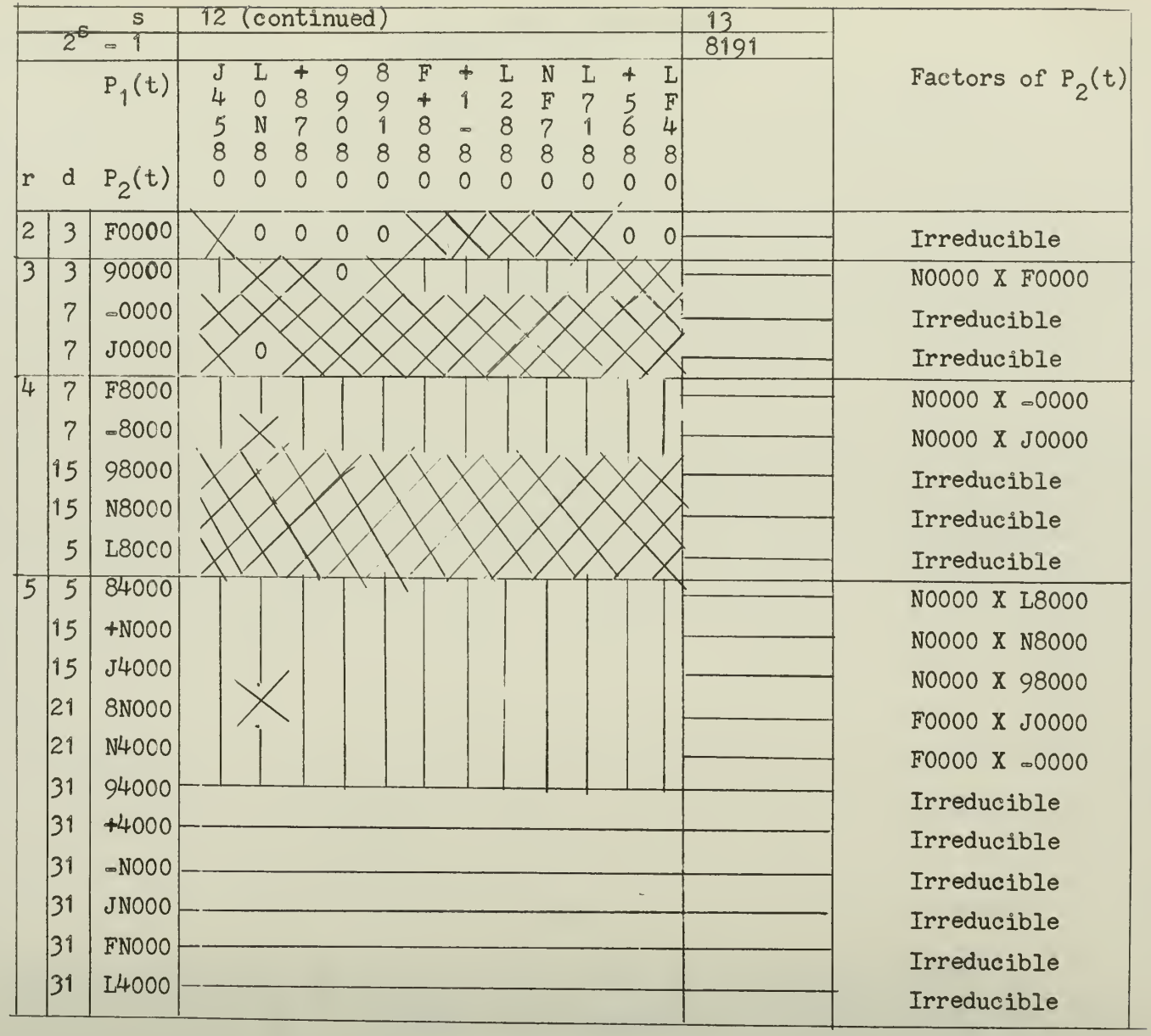
TABLE III-1 (continued)

s 12 (continued)

$2^s - 1$

r	d	$P_2(t)$	$P_1(t)$
			8 L J + N F N 8 8 + F 8 F J L - 9 - 9 9 F F + + + - J N J - 8
			- 3 J F 8 8 L 2 9 N N - 2 8 2 N 5 + J 9 9 1 0 2 1 5 - F 4 3 J J
			5 3 L - J 3 3 9 J 9 4 0 6 L 2 4 N 1 1 N 1 2 F 8 3 + 0 4 4 L 9
			8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2	3	F0000	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3	3	90000	0 0 0 0 0 0 0 0 0 0
	7	-0000	
	7	J0000	0
4	7	F8000	
	7	-8000	
	15	98000	
	15	N8000	
	5	L8000	
5	5	84000	
	15	+N000	
	15	J4000	
	21	8N000	
	21	N4000	
	31	94000	
	31	+4000	
	31	-N000	
	31	JN000	
	31	FN000	
	31	L4000	

TABLE III-1 (continued)



$2^s - 1$	s	$P_1(t)$	12 (continued)	13 8191	Factors of $P_2(t)$
			J L + 9 8 F + L N L + L 4 0 8 9 9 + 1 2 F 7 5 F 5 N 7 0 1 8 - 8 7 1 6 4 8 8 8 8 8 8 8 8 8 8 8 8		
r	d	$P_2(t)$	0 0 0 0 0 0 0 0 0 0 0 0		
2	3	F0000	0 0 0 0 0 0		Irreducible
3	3	90000	0		N0000 X F0000
	7	-0000			Irreducible
	7	J0000	0		Irreducible
4	7	F8000			N0000 X -0000
	7	-8000			N0000 X J0000
	15	98000			Irreducible
	15	N8000			Irreducible
	5	L8000			Irreducible
5	5	84000			N0000 X L8000
	15	+N000			N0000 X N8000
	15	J4000			N0000 X 98000
	21	8N000			F0000 X J0000
	21	N4000			F0000 X -0000
	31	94000			Irreducible
	31	+4000			Irreducible
	31	-N000			Irreducible
	31	JN000			Irreducible
	31	FN000			Irreducible
	31	L4000			Irreducible

TABLE III-1 (continued)

CHAPTER IV

TWO- AND THREE-DIMENSIONAL CODES

The two- and three-dimensional rasters are considerably more complicated and therefore more interesting, than the one-dimensional raster because a variety of different shapes are possible for these rasters. It should be noted that the method given below for testing whether or not a given polynomial generated a given spot-correcting code is a systematic one and can be easily programmed for use on a high-speed computer, even though it is explained here by means of examples.

Section IV-1. Diameter of a Two-Dimensional Raster.

Let us first consider how the diameter of the square two-dimensional raster, i.e., a raster whose configuration matrix L is the identity matrix, varies as a function of the alignment matrix M.

A two-dimensional alignment matrix is of the form $\begin{bmatrix} n & k_{12} \\ 0 & 1 \end{bmatrix}$, where n is the length of the raster and $0 \leq k_{12} \leq n - 1$. It is obvious that if the configuration matrix helping to determine a raster R is the identity matrix, then e^2, the square of the diameter of R, must either be a perfect square or the sum of two squares; thus we have $e^2 = 1, 2, 4, 5, 8, 9, 10, 13, 16, 17,$ etc. The following table, which was worked out by hand, gives the value of e^2 for each choice of n such that $1 \leq n \leq 17$ and $0 \leq k_{12} \leq n$.

n \ k_{12}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	1																
2	1	2	1															
3	1	2	2	1														
4	1	2	4	2	1													
5	1	2	5	5	2	1												
6	1	2	5	4	5	2	1											
7	1	2	5	5	5	5	2	1										
8	1	2	5	8	4	8	5	2	1									
9	1	2	5	9	5	5	9	5	2	1								
10	1	2	5	10	8	4	8	10	5	2	1							
11	1	2	5	10	10	5	5	10	10	5	2	1						
12	1	2	5	10	9	8	4	8	9	10	5	2	1					
13	1	2	5	10	10	13	5	5	13	10	10	5	2	1				
14	1	2	5	10	13	10	8	4	8	10	13	10	5	2	1			
15	1	2	5	10	17	9	13	5	5	13	9	17	10	5	2	1		
16	1	2	5	10	16	10	13	8	4	8	13	10	16	10	5	2	1	
17	1	2	5	10	17	13	10	13	5	5	13	10	13	17	10	5	2	1

TABLE IV-1

An examination of this table shows that for every n except n = 15, the maximum value of e^2 is given by the largest sum of squares which is less than or equal to n. In the case n = 15, k_{12} = 4 or 11, we have e^2 = 17, which is quite close to the theoretical maximum value $(2 \times 15)/\sqrt{3} = 10\sqrt{3} = 17.32\ldots$ given in Section II-2.

Similar tables could be worked out for any raster with a known configuration matrix.

We know by Theorem II-2-11 that $(t^{n_2} + 1)\, P_1(t)$, for certain $P_1(t)$, generates a code which corrects all spot-errors of diameter $< e_2/2$ on a raster of length $n = \mathrm{lcm}(n_1^*, n_2)$. The following example shows the application of this result to a typical raster.

Let R be the two-dimensional raster determined by configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ (the identity matrix) and alignment matrix $M = M_2M_1 = \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix} M_1$.

Table IV-1 shows that the raster R_2 determined by (L, M_2) has diameter $e_2 = \sqrt{5} = 2.236\ldots$; which means that $(t^6 + 1)\, P_1(t)$ generates a code which corrects all spot-errors of diameter $< e_2/2 = 1.118\ldots$, and, in particular that

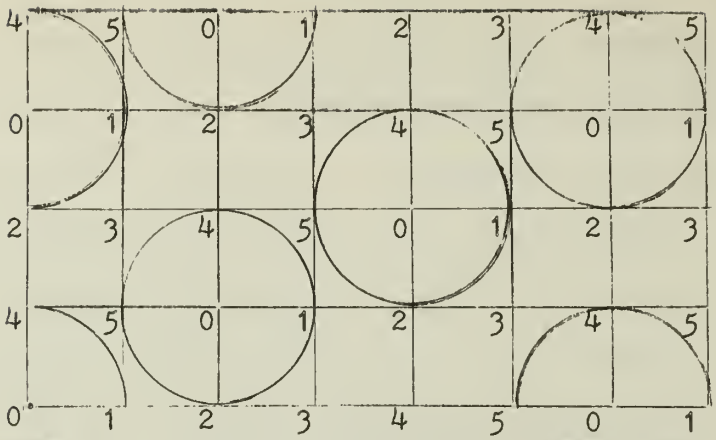
$(t^6 + 1)\ P_1(t)$ generates a spot-correcting code of diameter 1. From Figure IV-1, it is obvious that no spot of diameter ≤ 1 containing t^0 can have a point in common with any spot of diameter ≤ 1 containing t^{6q} for any integer q because each spot of diameter ≤ 1 is by definition wholly contained in each circle of radius 1 with center at one of its points. Figure IV-1 shows that each of these circles is disjoint from all the others, and therefore that $(t^6 + 1)\ P_1(t)$ generates a spot-correcting code of diameter 1 on R.

The fact that $(t^6 + 1)\ P_1(t)$ actually corrects all spot-errors of diameter $\leq 1.118\ldots$ is only of theoretical interest in this case, since there are no actual spots of diameter m such that $1 \leq m \leq 1.118\ldots$.

The following examples show that 1) there exists a raster R on which $(t^{n_2} + 1)\ P_1(t)$ fails to generate a code of diameter $e_2/2$, and 2) there exists a raster R on which $(t^{n_2} + 1)\ P_1(t)$ generates a code of diameter $e_2 - \epsilon$, where ϵ is any arbitrarily small positive number. Thus, the actual diameter of correction m varies from raster to raster within the absolute limits $e_2/2 \leq m \leq e_2$.

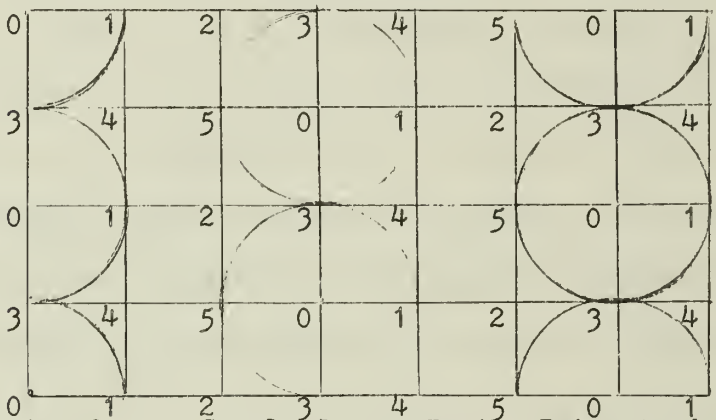
Let R be the two-dimensional raster determined by configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ as before and alignment matrix $M = M_2M_1 = \begin{bmatrix} 6 & 3 \\ 0 & 1 \end{bmatrix} M_1$. Table IV-1 shows that the raster R_2 determined by (L, M) has diameter $e_2 = \sqrt{4} = 2$, which means that $(t^6 + 1)\ P_1(t)$ generates a code which corrects all spot-errors of diameter $< e_2/2 = 1$, yet it can be shown that, for example, $(t^6 + 1)$ $(t^3 + t + 1) = t^9 + t^7 + t^6 + t^3 + t + 1$ (= -2N in sexadecimal notation) fails to generate a spot-correcting code of diameter 1 on a simple square raster of length 42 with $b_1 = 3$. From Figure IV-2, it is obvious that two distinct spots, one containing the point t^0 and the other containing the point t^{6q}, can have a third point in common.

On the other hand, let ϵ be any arbitrarily small positive number, let n_2 be any integer such that $n_2 \geq 3$, let R be the two-dimensional raster determined



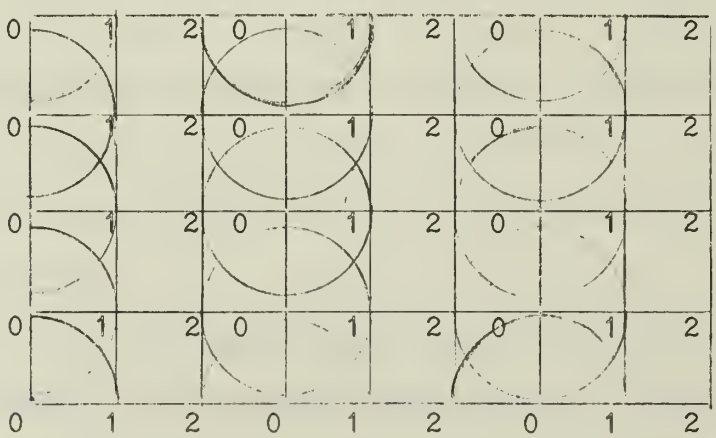
Circles of Diameter 2 on a Simple Square Raster Determined by $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $M = \begin{bmatrix} 6 & 2 \\ 0 & 1 \end{bmatrix}$ M_1. Each Point is Identified by its Residue Modulo 6.

Figure IV-1



Circles of Diameter 2 on a Simple Square Raster Determined by $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $M = \begin{bmatrix} 6 & 3 \\ 0 & 1 \end{bmatrix}$ M_1. Each Point is Identified by its Residue Modulo 6.

Figure IV-2



Circles of Diameter $2(1-\epsilon)$ on a Simple Square Raster Determined by $L = \begin{bmatrix} 1-\epsilon & 0 \\ 0 & 1 \end{bmatrix}$ and $M = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}$ M_1. Each Point is Identified by its Residue Modulo 3.

Figure IV-3

by configuration matrix $L = \begin{bmatrix} 1 & -\epsilon & 0 \\ 0 & & 1 \end{bmatrix}$ and alignment matrix $\begin{bmatrix} n_2 & 0 \\ 0 & 1 \end{bmatrix} M_1$. Then $(t^{n_2} + 1)\, P_1(t)$ generates a spot-correcting code of diameter $1 - \epsilon$ on R. From Figure IV-3, it is obvious that no spot of diameter $\leq 1 - \epsilon$ containing t^0 can have a point in common with any spot of diameter $\leq 1 - \epsilon$ containing t^{qn_2} for any integer q.

Most of the rasters studied in this paper are simple rasters, as defined in Definition II-3-3. However, there exist raster pairs R_2 and R such that the R raster is not simple and yet R_2 and R together satisfy the hypotheses of Theorem II-2-7. Since our most important theoretical results, Theorems II-2-11 and II-2-12, concern raster pairs related as in Theorem II-2-7, let us now consider in detail the construction of a raster R from a given raster R_2.

Let us take the raster R_2 to be a simple square raster of length n_2 and width b_1, so that $M_2 = \begin{bmatrix} n_2 & n_2 - b_1 \\ 0 & 1 \end{bmatrix}$. Suppose also that $M_1 = \begin{bmatrix} n_1 & h_{12} \\ 0 & 1 \end{bmatrix}$; then $M = M_2 M_1 = \begin{bmatrix} n_2 n_1 & h_{12} n_2 + (n_2 - b_1) \\ 0 & 1 \end{bmatrix}$.

As may be easily verified, the raster R corresponding to M is of length $n = n_2 n_1$ and width $b_1^* = b_1 + (n_1 - 1 - h_{12})\, n_2$. For example, if $M_2 = \begin{bmatrix} 5 & 3 \\ 0 & 1 \end{bmatrix}$, so that R_2 is of length 5 and width 2, (see comment following Definition II-2-4) and $M_1 = \begin{bmatrix} 10 & 6 \\ 0 & 1 \end{bmatrix}$, then $M = \begin{bmatrix} 50 & 33 \\ 0 & 1 \end{bmatrix}$, so that R is a simple square raster of length 50 and width 17. These rasters are respectively

$R_2 =$

4		
2	3	
0	1	

and

R =

34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

If for any M_1 it turns out that b_1^* does not itself divide n but there exists a d such that b_1^*/d does divide n, then the raster R determined by

(L, M_2M_1) may be thought of as an "interlinear" raster of length n and width b_1*/d. For example, if $M_2 = \begin{bmatrix} 5 & 4 \\ 0 & 1 \end{bmatrix}$ and $M_1 = \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix}$, then $M = \begin{bmatrix} 15 & 9 \\ 0 & 1 \end{bmatrix}$, so that R may be thought of as a simple square raster of length 15 and width 6. Since $b_1* = 6$ does not divide 15, but $b_1*/d = 6/2 = 3$ does divide 15; therefore we may think of R as an interlinear raster of length 15 and width 3. These rasters are respectively:

$R_2 = \begin{matrix} 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{matrix}$; simple R = $\begin{matrix} 12 & 13 & 14 & & & \\ 6 & 7 & 8 & 9 & 10 & 11 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{matrix}$;

and interlinear R = $\begin{matrix} 9 & 10 & 11 \\ 3 & 4 & 5 \\ 12 & 13 & 14 \\ 6 & 7 & 8 \\ 0 & 1 & 2 \end{matrix}$.

The simple raster is the easiest to implement physically because any mechanism that simply scans the raster from left to right and bottom to top (in this case, rows 1, 2, 3, 4, 5 in order) will obtain the information bits in their correct sequence on such a raster, whereas an "interlinear" raster would have to be scanned interlinearly (in this case, rows 1, 4, 2, 5, 3 in order) in order to obtain the bits in their correct sequence. On the other hand, any two simple rasters R derived from a given R_2 must either be of the same width b_1* or differ in width by a multiple of n_2, which may be inconvenient for some applications. This limitation is relaxed for the interlinear raster to the extent that rasters of width b_1*/d, for certain d, can also exist, thus giving us a wider choice of raster widths for a given n_2.

Section IV-2. The Direct Method.

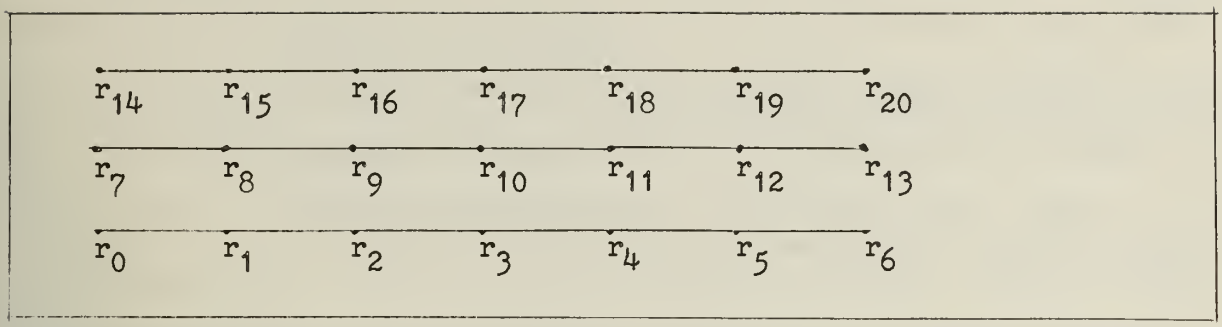
Now let us consider the problem of actually finding a polynomial P(t) which generates a spot-correcting code of given diameter a on a two-dimensional raster

of given length b. The first method which we shall discuss is the Direct Method described in Section II-3. Suppose that we are given a two-dimensional simple square raster of width b_1 and height b_2, so that the length of the raster is $b = b_2 b_1$. The set of normal polynomials of all spot-errors of diameter $\leq a$ is first determined either by inspection or by calculation, and a polynomial P(t) is chosen whose period is such that it is capable of generating the desired code. Then the sequence of calculations described at the end of Section I-2 is undertaken to determine whether or not P(t) does indeed generate a code which corrects the set of errors under consideration.

Let us suppose, for example, that we have been asked to find a polynomial which generates a spot-correcting code of diameter 1 on a simple square raster of width b_1 and height b_2, and thus of length $b = b_2 b_1$. The number of non-zero normal forms is three--namely 1, (single error) (.), $t + 1$ (horizontal double error) (..) and $t^{b_1} + 1$ (vertical double error) (:). We therefore must choose a polynomial P(t) of degree m such that $2^m - 1 \geq 3b$ since N(D)=1, the number of non-zero errors to be corrected, is 3b and the number of non-zero residues available with a polynomial of degree m is $2^m - 1$. Having chosen a likely polynomial, we calculate the order n of the polynomial 1 with respect to P(t) and then test whether or not $b \leq n$, because if $b > n$, P(t) obviously cannot generate a code of length b. If $b < n$, we will have a shortened cyclic code; if $b = n$, we will have a full-length cyclic code. If $b = n = (2^m - 1)/3$, the code will be close-packed and maximally efficient; if $b = n > (2^m - 1)/6$, the code will be

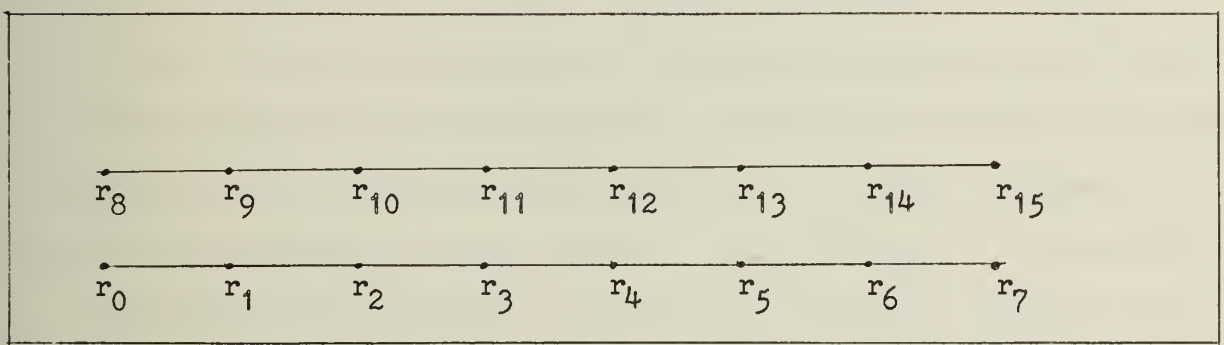
efficient. We next test whether or not $n \leq (2^m - 1)/3$, because if $n > (2^m - 1)/3$, P(t) obviously cannot generate a code with three separate cycles of length n. Assuming that $b \leq n \leq (2^m - 1)/3$, we proceed to find the representative polynomial (see Definition I-2-10) of $t + 1$ with respect to P(t) and its order with respect to P(t). If the representative polynomial of $t + 1$ with respect to P(t) is of order n and is not equal to 1 (which is of course its own representative polynomial), we proceed; otherwise, we conclude that P(t) does not generate the desired code. Assuming that the test of $t + 1$ has succeeded, we next proceed to find the representative polynomial of $t^{b_1} + 1$ with respect to P(t) and its order with respect to P(t). If the representative polynomial of $t^{b_1} + 1$ is of order n and is not equal to 1 or to the representative polynomial of $t + 1$, we then conclude that P(t) generates a spot-correcting code of diameter 1 on R; otherwise, we conclude that P(t) does not generate the desired code.

To see exactly how the Direct Method works, let us test whether or not $P(t) = t^6 + t^4 + t^2 + t + 1 = +F$ (see explanation of sexadecimal notation in Section III-2) generates a spot-correcting code of diameter 1 with $b_1 = 7$ (see Figure IV-4). First, we calculate the cycle containing the polynomial 1. This is done by forming the products $t(1) = t$, $t(t) = t^2$, $t(t^2) = t^3$, $t(t^3) = t^4$, $t(t^4) = t^5$, $t(t^5) = t^6 \equiv t^4 + t^2 + t + 1 \pmod{+F}$, $t(t^4 + t^2 + t + 1) = t^5 + t^3 + t^2 + t$, $t(t^5 + t^3 + t^2 + t) = t^6 + t^4 + t^3 + t^2 \equiv t^3 + t + 1 \pmod{+F}$, etc. In sexadecimal notation, we have $t^0 \equiv 001$, $t^1 \equiv 002$, $t^2 \equiv 004$, $t^3 \equiv 008$, $t^4 \equiv 010$, $t^5 \equiv 020$, $t^6 \equiv 017$, $t^7 \equiv 02F$, $t^8 \equiv 00-$, $t^9 \equiv 016$, $t^{10} \equiv 02N$, $t^{11} \equiv 00L$, $t^{12} \equiv 01F$, $t^{13} \equiv 03N$, $t^{14} \equiv 026$, $t^{15} \equiv 009$, $t^{16} \equiv 012$, $t^{17} \equiv 024$, $t^{18} \equiv 01L$, $t^{19} \equiv 03F$, $t^{20} \equiv 02-$, (all mod +F), which gives us finally 001, 002, 004, 008, 010, 020, 017, 02F, 00-, 016, 02N, 00L, 01F, 03N, 026, 009, 012, 024, 01L, 03F, 02-. Thus the first cycle is of length 21 and has representative polynomial $1 \equiv 001$. Next, we calculate the cycle containing



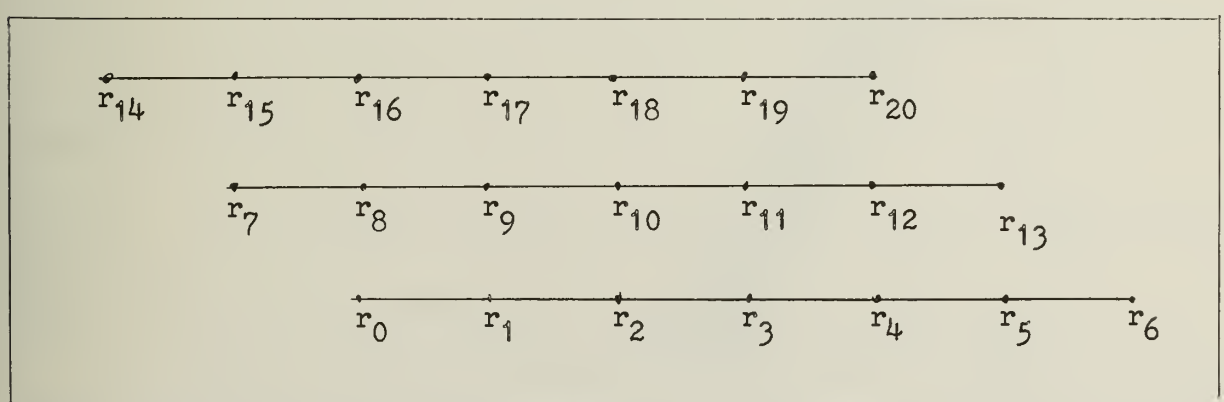
Simple Square Raster of Length 21 with $b_1 = 7$.

Figure IV-4



Simple Square Raster of Length 16 with $b_1 = 8$.

Figure IV-5



Offset Square Raster of Length 21 with $b_1 = 8$.

Figure IV-6

the polynomial $t + 1 = 002 + 001 = 003$. This gives us: 003, 006, 00N, 018, 030, 037, 039, 025, 01J, 03+, 023, 011, 022, 013, 026, 01-, 036, 03-, 021, 015, 02+. Thus this cycle is also of length 21 and has representative polynomial $t + 1 = 003$. Finally, we calculate the cycle containing the polynomial $t^7 + 1 = 02F + 001 = 02L$. This gives us 02L, 009, 012, 024, 01L, 03F, 02-, 001, 002, 004, 008, 010, 020, 017, 02F, 00-, 016, 02N, 00L, 01F, 03N. Thus this cycle is of length 21 and has representative polynomial $1 = 001$. Therefore, because 1 and $t^7 + 1$ have the same representative polynomial, $t^6 + t^4 + t^2 + t + 1 = +F$ does not generate the desired code.

The method just described is based on Theorem I-2-10 and would be used to investigate this problem on a high-speed computer such as Illiac. The following method, based on Theorem I-2-11, is superior for hand computation: We first determine the period of +F to be 21 by calculating the cycle containing the polynomial 1 as described above. The next step is to calculate the quotient $(t^{21} + 1)/(t^6 + t^4 + t^2 + t + 1) = t^{15} + t^{13} + t^{10} + t^7 + t^6 + t^3 + t + 1 = Q(t)$ as follows:

```
                  1010010011001011
          ┌──────────────────────
1010111   ) 10000000000000000001
            1010111
             101110 0
             1010111
                1011000
                1010111
                  1111000
                  1010111
                   1011110
                   1010111
                      1001000
                      1010111
                        1111100
                        1010111
                         1010111
```

Therefore, on a raster of length 21, we have:

$1 \times Q(t) = 00000101001001100101 1$; Normal polynomial = 000001010010011 001011

$(t + 1) \times Q(t) = 00000101001001100101 1$

$+ \underline{00001010010011001011 0}$

$00001111011010110111 01$; Normal polynomial = 000011110110101 011101

$(t^7 + 1) \times Q(t) = 00000101001001100101 1$

$+ \underline{10010011001011000001 0}$

$10010111000001010010 01$; Normal polynomial = 000001010010011 001011

Therefore, because Q(t) and $(t^7 + 1)Q(t)$ have the same normal polynomial, +F does not generate the desired code. This method is obviously easier than proceeding as the machine would, except for calculating the cycle containing the polynomial 1 and the initial division to determine Q(t).

A similar calculation shows that +F does generate a spot-correcting code of diameter 1 with $b_1 = 8$. Therefore, +F could be used to generate a shortened code of length 16 on the raster of Figure IV-5 or to generate a code of length 21 on the offset raster of Figure IV-6.

Tables IV-2 to IV-6 contain the results of an Illiac search for codes of various error-correcting capabilities generated by various polynomials. To illustrate the use of these tables, we note that the first line of Table IV-2 has the entries L8; 5; 1.000; 2. This means that the polynomial L8 = 11111000 = $t^4 + t^3 + t^2 + t + 1$ generates a spot-correcting code of diameter 1 on a simple square raster of length 5 with $b_1 = 2$ or 3; i.e., a raster determined by the matrix pair (L, M) with $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $M = \begin{bmatrix} 5 & 3 \\ 0 & 1 \end{bmatrix}$ or $\begin{bmatrix} 5 & 2 \\ 0 & 1 \end{bmatrix}$. The efficiency of either of these codes is 1.000. (Note that whenever a code exists for a given b_1, then it also exists for $n - b_1$.)

Corresponding to each non-symmetric polynomial P(t) of degree s in these tables, there is another polynomial P*(t) which may be obtained from P(t) by reversing the order of its coefficients (i.e., $P^*(t) = t^s P(t^{-1})$) and which generates a code with given length, diameter, and b_1 iff P(t) generates a code of

the given length, diameter, and b_1. This "mirror image" polynomial P*(t) is listed beside each non-symmetric P(t).

The intrinsic significance of these results is, of course, far outweighed by their value as building blocks in constructing codes of any desired length by the Combination Method.

P(t)	n	Efficiency	b_1
L8	5	1.000	2
84	5	.500	2
LN	6	.595	2
82	6	.297	2
8+,+2	14	.673	3,5
92	9	.438	2,4
9F,L2	15	.719	2,3,6,7
++	8	.391	3
+F,F+	21	1.000	2,8,10
-+	15	.719	2,7
J6	12	.579	2,5
LF	7	.344	2,3
81	7	.172	2,3
8-,J1	15	.359	2,3,5,6,7
9-,J9	35	.829	2,3,8,12,13,17
9L,L9	21	.500	2,3,6,7,8,9,10
-J	12	.289	2,4,5
NL,L3	14	.336	2,3,4,5,6
J-	9	.211	2,3,4
F7	15	.359	2,7
LL	8	.195	2,3
808	8	.097	2,3
828,+08	30	.356	3,5,7,9,11,13
888	12	.145	2,3,5
8+8,+88	14	.168	2,3,4,5,6
8-8,F88	15	.180	2,3,4,5,6,7

(TABLE IV-2 continued on next page)

P(t)	n	Efficiency	b_1
8J8,J88	51	.602	2,3,4,5,6,7,8,10,11,13,15,18,19,21, 22,23,24,25
938,F48	63	.742	2,3,6,8,9,10,12,15,18,21,23,24,27, 29,30,31
9+8,+N8	63	.742	2,3,6,8,9,11,12,13,15,18,19,21,24, 27,30,31
9N8	17	.203	2,3,4,5,6,7,8
9J8,JN8	42	.496	2,4,8,10,16,20
9L8,LN8	85	1.000	2,3,8,9,12,17,19,22,26,30,32,35,36, 37,42
+28	12	.145	3,5
+58,J28	21	.250	2,3,4,5,6,8,9,10
++8	10	.121	2,3,4
-58,J68	35	.415	2,3,8,12,13,17
-78,L68	30	.356	2,4,5,7,8,13,14
--8,FF8	85	1.000	2,8,9,14,19,26,29,30,31,32,34,35, 36,39,42
-F8	21	.250	2,4,5,10
N58,J18	85	1.000	2,5,6,7,8,11,17,20,23,24,27,28,32,41,42
N98	30	.356	2,4,7,8,13,14
NL8,L98	51	.602	2,4,5,6,7,8,9,10,12,13,14,15,18,19,20, 22,24,25
J58	24	.285	5,11
J-8,FJ8	42	.496	5,11,17,19
JJ8	15	.180	2,3,4,6,7
F38	12	.145	2,4,5
F-8	17	.203	2,3,4,5,6,7,8
L78	20	.238	3,7
LL8	9	.109	2,3,4

TABLE IV-2

Table IV-2 lists all polynomials of degree ≤ 8 with the property of generating a spot-correcting code of length n and diameter 1 on a simple two-dimensional square raster of width b_1 ; i.e., one with configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and alignment matrix $M = \begin{bmatrix} n & n-b_1 \\ 0 & 1 \end{bmatrix}$. That is, for each P(t) and b_1 appearing in the table, we have the polynomials 1 (.), t + 1 (..), and $t^{b_1} + 1$ (:) lying in separate cycles of length n with respect to P(t).

P(t)	n	Efficiency	b_1
LF	7	.673	2
81	7	.336	2
9L,L9	21	.993	8
NL,L3	14	.664	3,5
LL	8	.382	2
808	8	.191	2
888	12	.285	2
8+8,+88	14	.332	2,3,4,5
8-8,F88	15	.356	2,4,5,6
+58,J28	21	.496	4,8
++8	10	.238	2,3
JJ8	15	.356	3
LL8	9	.215	2,3

TABLE IV-3

Table IV-3 lists all polynomials of degree ≤ 8 with the property of generating a spot-correcting code of length n and diameter 1 on a simple two-dimensional hexagonal raster of width b_1; i.e., one with configuration matrix $L = \begin{bmatrix} 1 & -1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix}$ and alignment matrix $M = \begin{bmatrix} n & n-b_1 \\ 0 & 1 \end{bmatrix}$. That is, for each P(t) and b_1 appearing in the table, we have the polynomials 1 (.), t + 1 (..), $t^{b_1} + 1$ (°.), $t^{b_1+1} + 1$ (.°), $t^{b_1+1} + t + 1$ (.°.), $t^{b_1+1} + t^{b_1} + 1$ (°.°) lying in separate cycles of length n with respect to P(t).

P(t)	n	Efficiency	b_1
8+8,+88	14	.552	4
8-8,F88	15	.590	6
++8	10	.395	3
LL8	9	.355	3
804	9	.178	3
9NN,NF4	15	.295	6
--N,L74	21	.412	4,5,9
NJN,FNN	21	.412	4,5,7,9
LLN	10	.392	3

TABLE IV-4

Table IV-4 lists all polynomials of degree ≤ 9 with the property of generating a spot-correcting code of length n and diameter $\sqrt{2}$ on a simple two-dimensional square raster of width b_1; i.e., one with configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and alignment matrix $M = \begin{bmatrix} n & n-b_1 \\ 0 & 1 \end{bmatrix}$. That is, for each b_1 appearing in the table, we have the polynomials 1 (.), $t + 1$ (..), $t^{b_1 - 1} + 1$ ('.), $t^{b_1} + 1$ (:), $t^{b_1 + 1} + 1$ (.'), $t^{b_1} + t + 1$ (:.), $t^{b_1 + 1} + t + 1$ (.:), $t^{b_1} + t^{b_1 - 1} + 1$ (':), $t^{b_1 + 1} + t^{b_1} + 1$ (:'), and $t^{b_1 + 1} + t^{b_1} + t + 1$ (::) lying in separate cycles of length n with respect to P(t).

P(t)	n	Efficiency	b_1
8J9,9-1	31	.364	9
985,+19	21	.246	6
+2-,J45	31	.364	7
+F3,N75	23	.270	4,5,6
F3L,LN7	21	.246	4,5

TABLE IV-5

Table IV-5 lists all polynomials of degree ≤ 11 with the property of generating a spot-correcting code of length n and diameter 2 on a simple two-dimensional square raster of width b_1; i.e., one with configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and alignment matrix $M = \begin{bmatrix} n & n-b_1 \\ 0 & 1 \end{bmatrix}$. That is, for each b_1 appearing in the table, we have the polynomials 1 (.), $t + 1$ (..), $t^{b_1 - 1} + 1$ (·.), $t^{b_1} + 1$ (:), $t^{b_1 + 1} + 1$ (.·), $t^{b_1} + t + 1$ (:.), $t^{b_1 + 1} + t + 1$ (.:), $t^{b_1} + t^{b_1 - 1} + 1$ (·:), $t^{b_1 + 1} + t^{b_1} + 1$ (:·), $t^{b_1 + 1} + t^{b_1} + t + 1$ (::), $t^{2b_1} + t^{b_1 - 1} + 1$ (·.), $t^{b_1 + 1} + t^{b_1 - 1} + 1$ (·.·), $t^{2b_1} + t^{b_1 + 1} + t^{b_1 - 1} + 1$ (·.·), $t^{2b_1} + t^{b_1} + t^{b_1 - 1} + 1$ (·:), $t^{b_1 + 1} + t^{b_1} + t^{b_1 - 1} + 1$ (·:·), $t^{2b_1} + t^{b_1 + 1} + t^{b_1} + t^{b_1 - 1} + 1$ (·:·), $t^{2b_1} + 1$ (:), $t^{2b_1} + t^{b_1 + 1} + 1$ (.·), $t^{2b_1} + t^{b_1} + 1$ (:), $t^{2b_1} + t^{b_1 + 1} + t^{b_1} + 1$ (:·), $t^2 + 1$ (. .), $t^{b_1 + 1} + t^2 + 1$ (.·.), $t^2 + t + 1$ (...) and $t^{b_1 + 1} + t^2 + t + 1$ (.:.) lying in separate cycles of length n with respect to P(t).

P(t)	n	Efficiency	b_1	b_2
92	9	.579	$b_1 = 2$,	$b_2 = 2$
9F,L2	15	.954	$b_1 = 2$,	$b_2 = 3$
8-,J1	15	.477	$b_1 = 2$,	$b_2 = 3$
9L,L9	21	.664	$b_1 = 2$,	$b_2 = 3$
-J	12	.383	$b_1 = 2$,	$b_2 = 2$
NL,L3	14	.445	$b_1 = 3$,	$b_2 = 2$
J-	9	.289	$b_1 = 2$,	$b_2 = 2$

TABLE IV-6

Table IV-6 lists all polynomials of degree ≤ 7 which have the property of generating a spot-correcting code of length n and diameter 1 on a simple three-dimensional square raster of width b_1 and height b_2; i.e., one with configuration matrix $L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and alignment matrix $M = \begin{bmatrix} n & n-b_1 & n-b_1b_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. That is, for each b_1 and b_2 pair appearing in the table, we have the polynomials 1 (.), $t + 1$ (), $t^{b_1} + 1$ (), and $t^{b_1 b_2} + 1$ () lying in separate cycles of length n with respect to P(t).

Section IV-3. The Fire Method.

As we pointed out in Section II-3, the Fire Method may be thought of as dividing the error-correcting process into two parts--the first factor, $t^{n_2} + 1$, specifies the normal form of the error and $P_1(t)$ locates its position within the code word.

As an example of the use of the Fire Method, suppose we wish to find a polynomial of the form $P(t) = (t^{n_2} + 1)\, P_1(t)$ which generates a spot-correcting code of diameter a on a simple square raster (i.e., one with configuration matrix $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$) of width b_1 and height b_2 and thus of length $b = b_2 b_1$. We search Table IV-1 until we find a combination of n_2 and k_{12} for which we have diameter $e_2^2 > 4a^2$; i.e., $e_2/2 > a$, and such that $k_{12} = \text{Res}\,(n_2 - b_1) \pmod{n_2}$. Now let $P_1(t)$ be any polynomial such that $P_1(t)$ is prime to $t^{n_2} + 1$, has the property that $\text{lcm}(n_1, n_2) \geq b$, and such that each of 1, t+1, and $t^{b_1} + 1$ is of order n_1 with respect to $P_1(t)$. Then we know that $P(t) = (t^{n_2} + 1)P_1(t)$ generates a spot correcting code C of diameter a on the raster of length $n = \text{lcm}(n_1, n_2)$ determined by the matrix pair $(L, M) = (L, M_2 M_1)$, where $L = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $M_2 = \begin{bmatrix} n_2 & n_2 - b_1 \\ 0 & 1 \end{bmatrix}$, $M_1 = \begin{bmatrix} n_1 & n_1 - 1 \\ 0 & 1 \end{bmatrix}$, and $M = \begin{bmatrix} n_2 & n_2 - b_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} n_1 & n_1 - 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n & n - b_1 \\ 0 & 1 \end{bmatrix}$. If it happens that $n = b$, then the raster on which C is defined is of length b, as was desired. If $n > b$, we can still use C as a shortened cyclic code; that is, one which is formally of length n with respect to encoding and decoding processes but in which only b bits of the code are transmitted, the remainder being assumed to be zero and, of course, error-less.

To see exactly how the Fire Method works, suppose we wish to find a polynomial of the form $P(t) = (t^{n_2} + 1)\, P_1(t)$ which generates a spot-correcting code of diameter 1 on a simple square raster of width 7 and height 15, and thus

of length 105. Since Res(5-7)(mod 5) = 3, we see by Table IV-1 that the first n_2, k_{12} combination we may use is the combination $n_2 = 5$, $k_{12} = 3$ for which we have diameter $e_2^2 = 5$; i.e., $e_2/2 > 1$, as required. Now we must choose a $P_1(t)$ such that $P_1(t)$ is prime to $t^{n_2} + 1$, has the property that $\text{lcm}(n_1, n_2) \geq 105$, and each of the polynomials 1, $t + 1$, and $t^7 + 1$ is of order n_1 with respect to $P_1(t)$. Since we are looking for as efficient a code as possible, whether full-length or shortened, we know that $P_1(t)$ must be some polynomial of least degree and of period $n_1 \geq 105/5 = 21$; therefore, we choose $P_1(t)$ to be any primitive polynomial of degree 5 and thus of period $2^5 - 1 = 31$. Then $P(t) = (t^5 + 1)\, P_1(t)$ generates a spot-correcting code of length $5 \times 31 = 155$ on a raster of width 7; or, considering the code as a shortened cyclic code, a code of shortened length 105 on a raster of width 7 and height 15. The efficiency of the shortened code is given by $(ng + 1)/2^m = ((105)\ 3 + 1)/2^{10} = 316/1024 = .308$.

Section IV-4. The Combination Method.

The Combination Method is very important because it enables us to extend greatly the usefulness of code-generating polynomials found by the Direct Method. The Combination Method is similar to the Fire Method in that the final code-generating polynomial arrived at is of the form $P(t) = P_2(t)\, P_1(t)$ where $P_2(t)$ defines the normal form of the error to be corrected and $P_1(t)$ its location. The only real difference is that the factor polynomial $P_2(t)$ is arrived at by the Direct Method instead of being taken to be of the form $P_2(t) = t^{n_2} + 1$.

The steps to be followed in using the Combination Method are precisely the same as those for the Fire Method, except that instead of searching Table IV-1 for a $P_2(t)$ of the form $t^{n_2} + 1$ which generates a spot-correcting code of the

given diameter, we search in the appropriate one of Tables IV-2 through IV-6 for a $P_2(t)$ previously found by the Direct Method which generates a spot-correcting code of the given diameter on a suitable raster R.

To see exactly how the Combination Method works, let us solve the same problem as was solved by the Fire Method in Section IV-3. That is, suppose we wish to find a polynomial of the form $P(t) = P_2(t)P_1(t)$ which generates a spot-correcting code of diameter 1 on a simple square raster of width 7 and height 15, and thus of length 105.

We see from Table IV-2 that the polynomial $P_2(t) = L8000 = t^4 + t^3 + t^2 + t + 1$ generates a spot-correcting code of diameter 1 on a raster of length $n_2 = 5$ and $b_1 = 2$; therefore, as in the Fire Method, we choose $P_1(t)$ to be any primitive polynomial of degree 5 and thus of period $2^5 - 1 = 31$, which means that $P(t) = P_2(t)P_1(t)$ will be of degree $4 + 5 = 9$. Whereas the polynomial P(t) obtained by the Fire Method was of degree 10, and thus of efficiency $(3 \times 105 + 1)/1024 = .308$ the P(t) obtained by the Combination Method is of degree 9 and thus of efficiency $(3 \times 105 + 1)/512 = .617$. Since $.617 > \frac{1}{2}$, we know that the code obtained by the Combination Method is efficient and that therefore it would not be of any use to try to find a more efficient code. This code, like the one found by the Fire Method, is a spot-correcting code of length $5 \times 31 = 155$ on a raster of width 7, or a shortened cyclic code of length 105 on a raster of width 7 and height 15.

Since this code is efficient, there is no use in looking for a P(t) which generates a more efficient code, but it is possible to find a code for the 7 x 15 raster which is a full-length code instead of a shortened code. To do this, we note that in Table IV-2, -+ generates a spot-correcting code of diameter 1 on a raster of length 15 and width 7. Choosing $P_1(t)$ to be any primitive polynomial of degree 3 and thus of period $2^3 - 1 = 7$, we obtain a polynomial P(t)

which generates a spot-correcting code of diameter 1 on the 7 x 15 raster. Since P(t) is of degree 6 + 3 = 9, its efficiency is also .617.

Section IV-5. Comparison with Iterated Two-Dimensional Codes.

A number of previous investigations of cyclic group codes in two or more dimensions have been based on the concept of the iterated code; that is, a two-dimensional array with the property that each row of the array is a code word from some one-dimensional burst-error correcting code C_h of length n_h and of diameter e_h and, similarly, each column is a code word from some one-dimensional burst-error correcting code of length n_v and diameter e_v. The following analysis of these codes is due to Elspas.

It is clear that the product code $C_h \times C_v$ is the set of all $(n_h n_v)$-tuples, (f_{ij}), such that the polynomial $f(x,y) = \sum_{i=0}^{n_h-1} \sum_{j=0}^{n_v-1} f_{ij} x^i y^j$, is some multiple $A(x,y)P_h(x)P_v(y)$ of $P_h(x)P_v(y)$, where $P_h(x)$ and $P_v(y)$ are the generating polynomials of the cyclic group codes C_h and C_v respectively. It is also obvious that any error u which can be contained in the union of 1) a vertical band of width e_h and of any height, and 2) a horizontal band of height e_v and of any width, can be corrected by C_h and C_v independently and hence by $C_h \times C_v$. Elspas has discovered the following more remarkable result: $C_h \times C_v$ corrects any rectangular spot-error of size up to m_h digits wide and m_v digits high, where m_h and m_v are the degrees of $P_h(x)$ and $P_v(y)$ respectively.

To see what this result means, consider the burst-correcting code of length 15 and diameter 0 generated by any primitive polynomial of degree 4. Iterating this code with itself, we obtain a code of length 225 generated by a polynomial of degree 104 on which all spots of size 4 x 4 or less are

corrected, even though the codes C_h and C_v themselves correct only single errors.

To compare the results of this paper with Elspas's result, we note that according to the table following Definition II-2-7, for two-dimensional rasters we have $e_2^2 \leq (2/\sqrt{3})n_2$, which means $n_2 \geq 2\sqrt{3}(e_2/2)^2$. Since a size 4 x 4 spot on a square raster has diameter $3\sqrt{2}$, we know that the value of n_2 in the generating polynomial $P(t) = (t^{n_2} + 1)P_1(t)$ must be such that $n_2 \geq 2\sqrt{3}(3\sqrt{2})^2 = 36\sqrt{3} = 62.4$. Therefore, we construct a table similar to Table IV-1 beginning with $n_2 = 63$ and continuing until we find a combination of n_2 and k_{12} for which $e_2^2 > 72$; that is, $e_2/2 > 3\sqrt{2}$. Since we know that for the combination $n_2 = 81$ and $k_{12} = 9$ we have $e_2^2 = 81$, we know that there will certainly be an n_2 in the range $63 \leq n_2 \leq 81$ for which $e_2^2 > 72$. Then, choosing an appropriate $P_1(t)$, we obtain a polynomial of the form $P(t) = (t^{n_2} + 1)P_1(t)$ which generates a spot-correcting code of length at least 225 and diameter $\geq 3\sqrt{2}$. Since we know that $P_2(t)$ can be of degree ≤ 81 and $P_1(t)$ can be of degree ≤ 3 because $\text{lcm}(2^3 - 1, 81) > 225$, we see that the code obtained by our method is more efficient than Elspas's. For even greater efficiency, we could use the Direct Method to find a polynomial $P_2(t)$ which generates a spot-correcting code of diameter $3\sqrt{2}$, and then multiply by appropriate $P_1(t)$, if necessary, to give a code of length at least 225.

Thus it appears that, in this case, Elspas's method is about as easy to use, but gives a less efficient code. For spots of diameter ≤ 2, on the other hand, the methods given in this paper would seem to be better because very efficient code-generating polynomials can be found relatively easily whereas Elspas's method would require the use of polynomials $P_h(x)$ and $P_v(y)$ of degree > 2 to achieve a product code $C_h \times C_v$ of reasonable length.

BIBLIOGRAPHY

1. Abramson, N. M. "A Class of Systematic Codes for Non-Independent Errors", I R E Trans. on Information Theory, Vol. IT-5, pp. 150 - 157, December, 1959.

2. Albert, A. A. Fundamental Concepts of Higher Algebra, University of Chicago Press, 1956.

3. Birkhoff, G. and MacLane, S. A Survey of Modern Algebra, Revised Edition, New York, MacMillan, 1954.

4. Cassels, J. W. S. An Introduction to the Geometry of Numbers, Berlin, Springer Verlag, 1959.

5. Elspas, B. "Design and Instrumentation of Error-Correcting Codes", Final Report, RADC-TDR-62-511, Stanford Research Institute, October, 1962.

6. Elspas, B. and Short, R. A. "A Note on Optimum Burst-Error-Correcting Codes", I R E Trans. on Information Theory, Vol. IT-8, pp. 39 - 42, January, 1962.

7. Fire, P. "A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors", Sylvania Report RSL-E-2, Sylvania Reconnaissance Systems Laboratory, Mountain View, California, 1959.

8. Foulk, C. R. "Some Properties of Maximally Efficient Cyclic Burst-Error-Correcting Codes and Results of a Computer Search for such Codes", University of Illinois Digital Computer Laboratory, File Number 375, June 12, 1961.

9. Gorog, E. "Some New Classes of Cyclic Codes Used for Burst-Error Correction", IBM J. Research and Development, Vol. 7, pp. 102 - 111, April, 1963.

10. Hamming, R. W. "Error-Detecting and Error-Correcting Codes", Bell System Tech. J., Vol. 29, pp. 147 - 160, 1950.

11. Meggitt, J. E. "Error-Correcting Codes for Correcting Bursts of Errors", IBM J. Research and Development, Vol. 4, pp. 329 - 334, July, 1960.

12. Melas, C. M. and Gorog, E. "A Note on Extending Certain Codes to Correct Error Bursts in Longer Messages", IBM J. Research and Development, Vol. 7, pp. 151 - 152, April, 1963.

13. Peterson, W. W. Error-Correcting Codes, Massachusetts Institute of Technology Press and John Wiley & Sons, New York, 1961.

14. Stone, J. J. "Multiple Burst Error Correction", Information and Control, Vol. 4, pp. 324 - 331, 1961.

VITA

Clinton Ross Foulk was born in Wichita, Kansas, on January 24, 1930, and was graduated from El Dorado Senior High School in El Dorado, Kansas, in May, 1947. He attended the University of Kansas as a Summerfield Scholar from September, 1947 to June, 1951, graduating with a Bachelor of Arts degree in Astronomy. He attended the University of Lund in Lund, Sweden, from September, 1951 to May, 1952. From July, 1952 to June, 1956, he served at Wright-Patterson Air Force Base as a member of the United States Air Force, leaving with the rank of Staff Sergeant. In September, 1956, he entered the University of Illinois, as a University Fellow in the Digital Computer Laboratory, receiving a Master of Arts degree in Mathematics in February, 1958. He was an assistant in the Digital Computer Laboratory from June, 1958 to June, 1962. He is a member of Phi Beta Kappa, Omicron Delta Kappa, Pi Mu Epsilon, the American Mathematical Society, Society for Industrial and Applied Mathematics, Association for Computing Machinery, and Sigma Xi.

JUN 20 1969

UNIVERSITY OF ILLINOIS-URBANA
3 0112 003202287